NIST Trustworthy and Responsible AI
NIST AI 100-2e2025

Adversarial Machine Learning

A Taxonomy and Terminology of Attacks and Mitigations

Apostol Vassilev
Alina Oprea
Alie Fordyce
Hyrum Anderson
Xander Davies
Maia Hamin

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AI.100-2e2025

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Trustworthy and Responsible AI
NIST AI 100-2e2025

Adversarial Machine Learning

A Taxonomy and Terminology of Attacks and Mitigations

Apostol Vassilev
Computer Security Division
Information Technology Laboratory

Alina Oprea
Northeastern University

Maia Hamin
U.S. AI Safety Institute
National Institute of Standards and Technology

Alie Fordyce
Hyrum Anderson
Cisco

Xander Davies
U.K. AI Security Institute

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AI.100-2e2025

March 2025

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

Publication History
Approved by the NIST Editorial Review Board on 2025-03-20

How to Cite this NIST Technical Series Publication:
Vassilev A, Oprea A, Fordyce A, Anderson H, Davies X, Hamin M (2025) Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. (National Institute of Standards and Technology, Gaithersburg, MD) NIST Trustworthy and Responsible AI, NIST AI 100-2e2025.
https://doi.org/10.6028/NIST.AI.100-2e2025

Author ORCID iDs
Apostol Vassilev: 0000-0002-9081-3042
Alina Oprea: 0000-0002-4979-5292
Maia Hamin: 0009-0009-3834-6553

Contact Information
ai-100-2@nist.gov

Additional Information
Additional information about this publication is available at https://csrc.nist.gov/pubs/ai/100/2/e2025/final, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

This NIST Trustworthy and Responsible AI report provides a taxonomy of concepts and defines terminology in the field of adversarial machine learning (AML). The taxonomy is arranged in a conceptual hierarchy that includes key types of ML methods, life cycle stages of attack, and attacker goals, objectives, capabilities, and knowledge. This report also identifies current challenges in the life cycle of AI systems and describes corresponding methods for mitigating and managing the consequences of those attacks. The terminology used in this report is consistent with the literature on AML and is complemented by a glossary of key terms associated with the security of AI systems. Taken together, the taxonomy and terminology are meant to inform other standards and future practice guides for assessing and managing the security of AI systems by establishing a common language for the rapidly developing AML landscape.

Keywords

artificial intelligence; machine learning; attack taxonomy; abuse; data poisoning; evasion; privacy breach; attack mitigation; large language model; chatbot.

NIST Trustworthy and Responsible AI

The National Institute of Standards and Technology (NIST) promotes U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life. Among its broad range of activities, NIST contributes to the research, standards, evaluations, and data required to advance the development, use, and assurance of trustworthy artificial intelligence (AI).

Table of Contents

Audience . . . viii
Background . . . viii
Trademark Information . . . viii
How to Read This Document . . . ix
Acknowledgments . . . ix
Author Contributions . . . ix
Predictive AI and Generative AI Taxonomy Index . . . x
Executive Summary . . . xii
1. Introduction . . . 1
2. Predictive AI Taxonomy . . . 4
2.1. Attack Classification . . . 4
2.1.1. Stages of Learning . . . 5
2.1.2. Attacker Goals and Objectives . . . 6
2.1.3. Attacker Capabilities . . . 7
2.1.4. Attacker Knowledge . . . 8
2.1.5. Data Modality . . . 9
2.2. Evasion Attacks and Mitigations . . . 11
2.2.1. White-Box Evasion Attacks . . . 12
2.2.2. Black-Box Evasion Attacks . . . 15
2.2.3. Transferability of Attacks . . . 15
2.2.4. Evasion attacks in the real world . . . 16
2.2.5. Mitigations . . . 17
2.3. Poisoning Attacks and Mitigations . . . 19
2.3.1. Availability Poisoning . . . 19
2.3.2. Targeted Poisoning . . . 21
2.3.3. Backdoor Poisoning . . . 22
2.3.4. Model Poisoning . . . 26
2.3.5. Poisoning Attacks in the Real World . . . 27
2.4. Privacy Attacks and Mitigations . . . 28
2.4.1. Data Reconstruction . . . 28

2.4.2. Membership Inference 29
2.4.3. Property Inference 30
2.4.4. Model Extraction 31
2.4.5. Mitigations 32
3. Generative AI Taxonomy 34
3.1. Attack Classification 34
3.1.1. GenAI Stages of Learning 36
3.1.2. Attacker Goals and Objectives 39
3.1.3. Attacker Capabilities 40
3.2. Supply Chain Attacks and Mitigations 41
3.2.1. Data Poisoning Attacks 42
3.2.2. Model Poisoning Attacks 42
3.2.3. Mitigations 42
3.3. Direct Prompting Attacks and Mitigations 43
3.3.1. Attack Techniques 44
3.3.2. Information Extraction 46
3.3.3. Mitigations 48
3.4. Indirect Prompt Injection Attacks and Mitigations 50
3.4.1. Availability Attacks 51
3.4.2. Integrity Attacks 51
3.4.3. Privacy Compromise 52
3.4.4. Mitigations 53
3.5. Security of Agents 54
3.6. Benchmarks for AML Vulnerabilities 54
4. Key Challenges and Discussion 55
4.1. Key Challenges in AML 55
4.1.1. Trade-Offs Between the Attributes of Trustworthy AI 55
4.1.2. Theoretical Limitations on Adversarial Robustness 56
4.1.3. Evaluation 57
4.2. Discussion 57
4.2.1. The Scale Challenge 57

4.2.2. Supply Chain Challenges 58
4.2.3. Multimodal Models 58
4.2.4. Quantized Models 59
4.2.5. Risk Management in Light of AML 59
4.2.6. AML and Other AI System Characteristics 60
Appendix: Glossary 107

List of Figures

Figure 1. **Taxonomy of attacks on PredAI systems** 4
Figure 2. **Taxonomy of attacks on GenAI systems** 35
Figure 3. **Example LLM Training Pipeline used for InstructGPT [281]** 36
Figure 4. **LLM enterprise adoption pipeline** 37
Figure 5. **LLM enterprise adoption reference architecture** 38
Figure 6. **Retrieval-augmented generation** 39
Figure 7. **Map of the development and deployment life cycle of an AI model for broad-scale query access** 47
Figure 8. Pareto optimality 55

Audience

The intended primary audience for this document includes individuals and groups who are responsible for designing, developing, deploying, evaluating, and governing AI systems.

Background

This document is the result of an extensive literature review, conversations with experts in adversarial machine learning, and research performed by the authors in adversarial machine learning.

Trademark Information

All trademarks and registered trademarks belong to their respective organizations.

The Information Technology Laboratory (ITL) at NIST develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines.

This NIST Trustworthy and Responsible AI report focuses on identifying, addressing, and managing risks associated with adversarial machine learning. While practical guidance[1] published by NIST may serve as an informative reference, this guidance remains voluntary.

The content of this document reflects recommended practices. This document is not intended to serve as or supersede existing regulations, laws, or other mandatory guidance.

[1] In the context of this paper, the terms "practice guide," "guide," "guidance," and the like are consensus-created informative references that are intended for voluntary use. They should not be interpreted as equal to the use of the term "guidance" in a legal or regulatory context. This document does not establish any legal standard or any other legal requirement or defense under any law, nor does it have the force or effect of law.

How to Read This Document

This document uses the terms "AI technology," "AI system," and "AI applications" interchangeably. Terms related to the machine learning pipeline, such as "ML model" or "algorithm," are also used interchangeably in this document. Depending on context, the term "system" may refer to the broader organizational and/or social ecosystem within which the technology was designed, developed, deployed, and used instead of the more traditional use related to computational hardware or software.

Important reading notes:

- This document includes a series of blue callout boxes that highlight nuances and important takeaways.
- This document contains links shown in blue. Clicking on them will bring the reader to the relevant resource. Links links in the References point to external sources.
- Terms that are used but not defined or explained in the text are listed and defined in the Glossary. They are displayed in small caps in the text. Clicking on a word shown in SMALL CAPS (e.g., ADVERSARIAL EXAMPLE) takes the reader directly to the definition of that term in the Glossary. From there, one may click on the page number shown at the end of the definition to return.
- This document provides an Index of attack types to easily navigate and reference attacks and corresponding mitigations.

Acknowledgments

The authors wish to thank all of the people and organizations who submitted comments on the draft version of this paper. The received comments and suggested references were essential to improving the document and the future direction of this work. The authors also want to thank the many NIST, U.S. AI Safety Institute, and U.K. AI Security Institute colleagues who assisted in updating this document.

Author Contributions

The authors contributed equally to this work.

Predictive AI and Generative AI Taxonomy Index

- **Predictive AI Attacks Taxonomy**
  - Availability Violations (ID: NISTAML.01)
    * Model Poisoning (ID: NISTAML.011)
    * Clean-label Poisoning (ID: NISTAML.012)
    * Data Poisoning (ID: NISTAML.013)
    * Energy-latency (ID: NISTAML.014)
  - Integrity Violations (ID: NISTAML.02)
    * Clean-label Poisoning (ID: NISTAML.012)
    * Clean-label Backdoor (ID: NISTAML.021)
    * Evasion (ID: NISTAML.022)
    * Backdoor Poisoning (ID: NISTAML.023)
    * Targeted Poisoning (ID: NISTAML.024)
    * Black-box Evasion (ID: NISTAML.025)
    * Model Poisoning (ID: NISTAML.026)
  - Privacy Compromises (ID: NISTAML.03)
    * Model Extraction (ID: NISTAML.031)
    * Reconstruction (ID: NISTAML.032)
    * Membership Inference (ID: NISTAML.033)
    * Property Inference (ID: NISTAML.034)
  - Supply Chain Attacks (ID: NISTAML.05)
    * Model Poisoning (ID: NISTAML.051)
- **Generative AI Attacks Taxonomy**
  - Availability Violations (ID: NISTAML.01)
    * Data Poisoning (ID: NISTAML.013)
    * Indirect Prompt Injection (ID: NISTAML.015)
    * Prompt Injection (ID: NISTAML.018)
  - Integrity Violations (ID: NISTAML.02)

    * Data Poisoning (ID: NISTAML.013)
    * Indirect Prompt Injection (ID: NISTAML.015)
    * Prompt Injection (ID: NISTAML.018)
    * Backdoor Poisoning (ID: NISTAML.023)
    * Targeted Poisoning (ID: NISTAML.024)
    * Misaligned Outputs (ID: NISTAML.027)
  - Privacy Compromises (ID: NISTAML.03)
    * Indirect Prompt Injection (ID: NISTAML.015)
    * Prompt Injection (ID: NISTAML.018)
    * Backdoor Poisoning (ID: NISTAML.023)
    * Membership Inference (ID: NISTAML.033)
    * Prompt Extraction (ID: NISTAML.035)
    * Leaking information from user interactions (ID: NISTAML.036)
    * Training Data Attacks (ID: NISTAML.037)
    * Data Extraction (ID: NISTAML.038)
    * Compromising connected resources (ID: NISTAML.039)
  - Misuse Violations (ID: NISTAML.04)
    * Prompt Injection (ID: NISTAML.018)
  - Supply Chain Attacks (ID: NISTAML.05)
    * Model Poisoning (ID: NISTAML.051)

Executive Summary

This NIST Trustworthy and Responsible AI report describes a taxonomy and terminology for ADVERSARIAL MACHINE LEARNING (AML) that may aid in securing applications of artificial intelligence (AI) against adversarial manipulations and attacks.

The statistical, data-based nature of ML systems opens up new potential vectors for attacks against these systems' security, privacy, and safety, beyond the threats faced by traditional software systems. These challenges span different phases of ML operations such as the potential for adversarial manipulation of training data; the provision of adversarial inputs to adversely affect the performance of the AI system; and even malicious manipulations, modifications, or interactions with models to exfiltrate sensitive information from the model's training data or to which the model has access. Such attacks have been demonstrated under real-world conditions, and their sophistication and impacts have been increasing steadily.

The field of AML is concerned with studying these attacks. It must consider the capabilities of attackers, the model or system properties that attackers might seek to violate in pursuit of their objectives, and the design of attack methods that exploit vulnerabilities during the development, training, and deployment phases of the ML life cycle. It is also concerned with the design of ML algorithms and systems that can withstand these security and privacy challenges, a property often known as robustness [274].

To taxonimize these attacks, this report differentiates between predictive and generative AI systems and the attacks relevant to each. It considers the components of an AI system including the data; the model itself; the processes for training, testing, and deploying the model; and the broader software and system contexts into which models may be embedded, such as cases where Generative Artificial Intelligence (GenAI) models are deployed with access to private data or equipped with tools to take actions with real-world consequences.

Thus, the attacks within this taxonomy are classified relative to: (i) the AI system type, (ii) the stage of the ML life cycle process in which the attack is mounted, (iii) the attacker's goals and objectives in terms of the system properties they seek to violate, (iv) the attacker's capabilities and access, and (v) the attacker's knowledge of the learning process and beyond.

This report adopts the concepts of security, resilience, and robustness of ML systems from the NIST AI Risk Management Framework. Security, resilience, and robustness are gauged by risk, which is a measure of the extent to which an entity (e.g., a system) is threatened by a potential circumstance or event (e.g., an attack) and the severity of the outcome should such an event occur. However, this report does not make recommendations on risk tolerance (i.e., the level of risk that is acceptable to organizations or society) because it is highly contextual and specific to applications and use cases.

The spectrum of effective attacks against ML is wide, rapidly evolving, and covers all phases of the ML lifecycle — from design and implementation to training, testing, and deployment in the real world. The nature and power of these attacks are different and their impacts may depend not only on the vulnerabilities of the ML models but also the weaknesses of the infrastructure in which the AI systems are deployed. AI system components may also be adversely affected by design and implementation flaws that cause failures outside the context of adversarial use, such as inaccuracy. However, these kinds of flaws are not within the scope of the literature on AML or the attacks in this report.

In addition to defining a taxonomy of attacks, this report provides corresponding methods for mitigating and managing the consequences of those attacks in the life cycle of AI systems, and outlines the limitations of widely used mitigation techniques to raise awareness and help organizations increase the efficacy of their AI risk-mitigation efforts. The terminology used in this report is consistent with the literature on AML and is complemented by a glossary that defines key terms associated with the field of AML in order to assist non-expert readers. Taken together, the taxonomy and terminology are meant to inform other standards and future practice guides for assessing and managing the security of AI systems by establishing a common language for the rapidly developing AML landscape. Like the taxonomy, the terminology and definitions are not intended to be exhaustive but rather to serve as a starting point for understanding and aligning on key concepts that have emerged in the AML literature.

1. Introduction

Artificial intelligence (AI) systems have been on a global expansion trajectory for several years [267]. These systems are being developed by and widely deployed into the economies of numerous countries, with increasing opportunities for people to use AI systems in many spheres of their lives [92]. This report distinguishes between two broad classes of AI systems: predictive AI (PredAI) and generative AI (GenAI). Although the majority of industrial applications of AI systems are still dominated by PredAI systems, there has been a recent increase in the adoption of GenAI systems in business and consumer contexts. As these systems permeate the digital economy and become essential parts of daily life, the need for their secure, robust, and resilient operation grows. These operational attributes are critical elements of trustworthy AI in the NIST AI Risk Management Framework [274] and the NCSC Machine Learning Principles [266].

The field of ADVERSARIAL MACHINE LEARNING (AML) studies attacks against ML systems that exploit the statistical, data-based nature of ML systems. Despite the significant progress of AI and machine learning (ML) in different application domains, these technologies remain vulnerable to attacks that can cause spectacular failures. The chances of these kinds of failure increase as ML systems are used in contexts where they may be subject to novel or adversarial interactions, and the consequences grow more dire as these systems are used in increasingly high-stakes domains. For example, in PredAI computer vision applications for object detection and classification, well-known cases of adversarial perturbations of input images have caused autonomous vehicles to swerve into lanes going in the opposite direction, stop signs to be misclassified as speed limit signs, and even people wearing glasses to be misidentified in high-security settings [121, 187, 332, 349]. Similarly, the potential for adversarial input to trick ML models into revealing hidden information has become more urgent as more ML models are being deployed in fields like medicine, where medical record leaks can expose sensitive personal information [25, 171].

In GenAI, large language models (LLMs) [13, 15, 49, 85, 102, 236, 247, 277, 279, 348, 365, 371, 372, 436] are increasingly becoming an integral part of software applications and internet infrastructure. LLMs are being used to create more powerful online search tools, help software developers write code, and power chatbots that are used by millions of people every day [255]. LLMs are also being augmented to create more useful AI systems, including through interactions with corporate databases and documents to enable powerful RETRIEVAL-AUGMENTED GENERATION (RAG) (RAG) [210] and through training- or inference-time techniques to enable LLMs to take real-world actions, such as browsing the web or using a bash terminal as an LLM-based AGENT [167, 261, 278, 419]. Thus, vulnerabilities in GenAI systems may expose a broad attack surface for threats to the privacy of sensitive user data or proprietary information about models' architecture or training data, and create risks to the integrity and availability of widely used systems.

As GenAI adoption has grown, the increasing capability of these systems has created another challenge for model developers: how to manage the risks created by unwanted or harmful uses of these systems' capabilities.[275] As model developers have increasingly sought to apply technical interventions to reduce models' potential for misuse, another surface for high-stakes AML attacks has emerged in attacks that attempt to circumvent or disrupt these protections.

Fundamentally, many AI systems are susceptible both to AML attacks and to attacks that more closely resemble traditional cybersecurity attacks, including attacks against the platforms on which they are deployed. This report focuses on the former and considers the latter to be within the scope of traditional cybersecurity taxonomies.

Both PredAI and GenAI systems are vulnerable to attacks enabled by a range of attacker capabilities throughout the development and deployment life cycle. Attackers can manipulate training data [327], including the Internet data used in large-scale model training [57], or can modify test-time inference data and resources by adding adversarial perturbations or suffixes. Attackers can also attack the components used to make AI systems by inserting TROJAN functionality. As organizations increasingly rely on pre-trained models that could be used directly or fine-tuned with new datasets to enable different tasks, their vulnerability to these attacks increases.

Modern cryptography often relies on algorithms that are secure in an information-theoretic sense, that is, those that can be formally proven to ensure security under certain conditions. However, there are no information-theoretic security proofs for the widely used ML algorithms in modern AI systems. Moreover, information-theoretic *impossibility* results that set limits on the effectiveness of widely used mitigation techniques have begun to appear in the literature [124, 140, 432]. As a result, many of the advances in developing mitigations against different classes of AML attacks tend to be empirical and limited in nature, adopted because they appear to work in practice rather than because they provide information-theoretic security guarantees. Thus, many of these mitigations may themselves be vulnerable to new discoveries and evolutions in attacker techniques.

This report offers guidance for the development of:

- Standardized terminology for AML terms that can be used across relevant ML and cybersecurity communities. There are notable differences in terminology in different stakeholder communities and it is important to work towards bridging the differences as AI is increasingly adopted throughout enterprise and consumer contexts.
- A taxonomy of the most widely studied and currently effective attacks in AML, including:
  - Evasion, poisoning, and privacy attacks for PredAI systems
  - Poisoning, direct prompting, and indirect prompt injection attacks for GenAI systems
- A discussion of potential mitigations for these attacks and the limitations of existing mitigation techniques

NIST intends to update this report as new developments emerge in AML attacks and mitigations.

> This report provides a categorization of common classes of attacks and their mitigations for PredAI and GenAI systems. This report is not intended to provide an exhaustive survey of all available literature on Adversarial ML, which includes more than 11,354 references on arXiv.org since 2021 as of July 2024.

This report is organized into three sections.

- Section 2 considers PredAI systems. Section 2.1 introduces the taxonomy of attacks for PredAI systems, which defines the broad categories of attacker objectives and goals, and identifies the capabilities that an adversary must leverage to achieve the corresponding objectives. Specific attack classes are also introduced for each type of capability. Sections 2.2, 2.3, and 2.4 discuss the major classes of attacks: evasion, poisoning, and privacy, respectively. A corresponding set of mitigations for each class of attacks is provided in the attack class sections.
- Section 3 considers GenAI systems. Section 3.1 introduces the taxonomy of attacks for GenAI systems and defines the broad categories of attacker objectives and adversary capabilities relevant to these systems. Specific attack classes are introduced for each type of capability, along with relevant mitigations.
- Section 4 discusses remaining challenges in the field, including limitations to widely used mitigation techniques. The intent is to raise awareness of open questions in the field of AML and to call attention to trends that may shape risk and risk management practices in future.

2. Predictive AI Taxonomy

2.1. Attack Classification

Figure 1 introduces a taxonomy of attacks in AML on PredAI systems, based on attacker goals and objectives, capabilities, and knowledge.

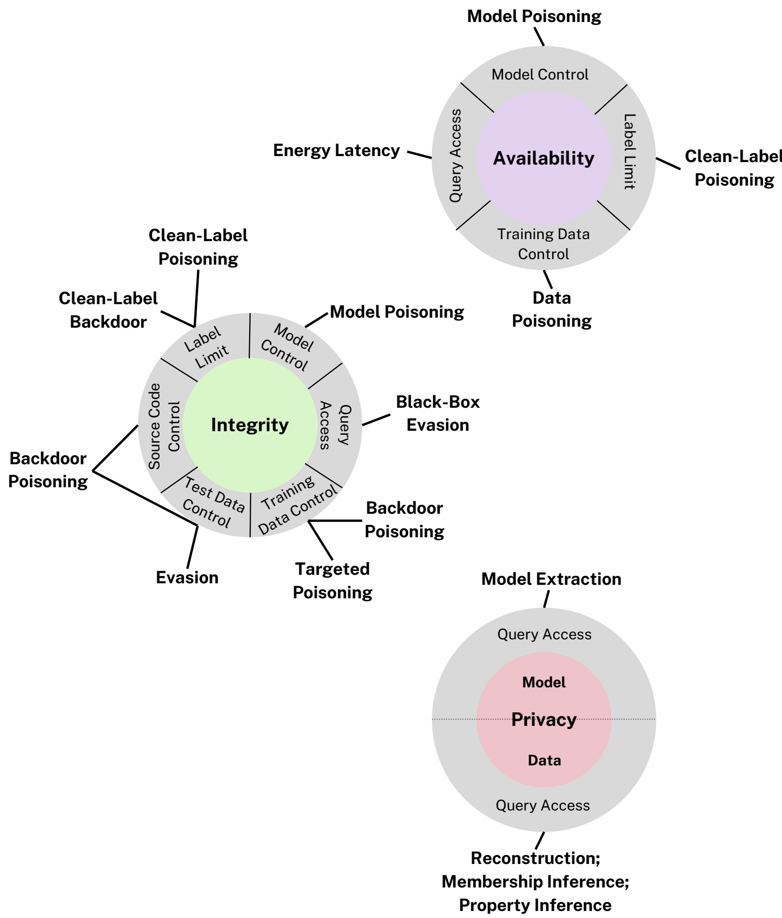

Figure 1. Taxonomy of attacks on PredAI systems

The attacker's objectives are shown as disjointed circles with the attacker's goal at the center of each circle: **availability** breakdown, **integrity** violation, and **privacy** compromise. The capabilities that an adversary must leverage to achieve their objectives are shown in the outer layer of the objective circles. Attack classes are shown as callouts connected to the capabilities required to mount each attack. Multiple attack classes that require the same capabilities to reach the same objective are shown in a single callout.

These attacks are classified according to the following dimensions: 1) learning method and stage of the learning process when the attack is mounted, 2) attacker goals and objectives,

3) attacker capabilities, and 4) attacker knowledge of the learning process. Several adversarial attack classification frameworks have been introduced in prior works [42, 358], and the goal here is to create a standard terminology for adversarial attacks on ML that unifies existing work.

2.1.1. Stages of Learning

Predictive machine learning involves a TRAINING STAGE in which a model is learned and a DEPLOYMENT STAGE in which the model is deployed on new, unlabeled data samples to generate predictions. In the case of SUPERVISED LEARNING, labeled training data is given as input to a training algorithm in the training stage, and the ML model is optimized to minimize a specific loss function. Validation and testing of the ML model is usually performed before the model is deployed in the real world. Common supervised learning techniques include CLASSIFICATION in which the predicted labels or *classes* are discrete and REGRESSION in which the predicted labels or *response variables* are continuous.

Other learning paradigms in the ML literature include UNSUPERVISED LEARNING, which trains models using unlabeled data at training time; SEMI-SUPERVISED LEARNING in which a small set of examples have labels, while the majority of samples are unlabeled; REINFORCEMENT LEARNING in which an agent interacts with an environment and learns an optimal policy to maximize its reward; FEDERATED LEARNING in which a set of clients jointly train an ML model by communicating with a server that performs an aggregation of model updates; and ENSEMBLE LEARNING, which is an approach that seeks better predictive performance by combining the predictions from multiple models.

Most PredAI models are DISCRIMINATIVE, i.e., learn only a decision boundary, such as LOGISTIC REGRESSION, SUPPORT VECTOR MACHINES, and CONVOLUTIONAL NEURAL NETWORKS. GenAI models may also be used in predictive tasks, such as sentiment analysis [125] .

AML literature predominantly considers adversarial attacks against AI systems that could occur at either the training stage or the deployment stage. During the training stage, the attacker might control part of the training data, their labels, the model parameters, or the code of ML algorithms, resulting in different types of poisoning attacks. During the deployment stage, the ML model is already trained, and the adversary could mount evasion attacks to create integrity violations and change the ML model's predictions, as well as privacy attacks to infer sensitive information about the training data or the ML model.

Training-time attacks. POISONING ATTACKS [40] occur during the ML training stage. In a DATA POISONING attack [40, 148], an adversary controls a subset of the training data by either inserting or modifying training samples. In a MODEL POISONING attack [222], the adversary controls the model and its parameters. Data poisoning attacks are applicable to all learning paradigms, while model poisoning attacks are most prevalent in federated learning [190], where clients send local model updates to the aggregating server, and in supply-chain attacks, where malicious code may be added to the model by suppliers of model technology.

Deployment-time attacks. Other types of attacks can be mounted against deployed models. Evasion attacks modify testing samples to create ADVERSARIAL EXAMPLE [38, 144, 362], which are similar to the original sample (e.g., according to certain distance metrics) but alter the model predictions to the attacker's choices. Other attacks, such as availability attacks and privacy attacks including membership inference [342] and data reconstruction [110], can also be mounted by attackers with query access to a deployed ML model.

2.1.2. Attacker Goals and Objectives

The attacker's objectives are classified along three dimensions according to the three main types of security violations considered when analyzing the security of a system: availability breakdown, integrity violation, and privacy compromise. Figure 1 separates attacks into three disjointed circles according to their objective, and the attacker's objective is shown at the center of each circle.

Availability breakdown [NISTAML.01] [Back to Index]. An AVAILABILITY BREAKDOWN attack is a deliberate interference with a PredAI system to disrupt the ability of other users or processes to obtain timely and reliable access to its services. This attack type may be initiated at training or deployment time, although its impacts are typically experienced at deployment time. Availability attacks can be mounted via data poisoning, when the attacker controls a fraction of the training set; via model poisoning, when the attacker controls the model parameters; or as ENERGY-LATENCY ATTACK via query access. Data poisoning availability attacks have been proposed for SUPPORT VECTOR MACHINES [40], linear regression [179], and even neural networks [228, 260], while model poisoning attacks have been designed for neural networks [222] and federated learning [22].

- **Energy latency attacks [NISTAML.014] [Back to Index].** Recently, ENERGY-LATENCY ATTACK, a type of availability attacks that require only black-box access to the model, have been developed for neural networks across many different tasks in computer vision and natural language processing (NLP) [345].

Integrity violation [NISTAML.02] [Back to Index]. An INTEGRITY VIOLATION attack is a deliberate interference with a PredAI system to force it to misperform against its intended objectives and produce predictions that align with the adversary's objective. An attacker can cause an integrity violation by mounting an evasion attack at deployment time or a poisoning attack at training time. Evasion attacks require the modification of testing samples to create adversarial examples that are misclassified by the model while often remaining stealthy and imperceptible to humans [38, 144, 362]. Integrity attacks via poisoning can be classified as TARGETED POISONING ATTACK [137, 330], BACKDOOR POISONING ATTACK [148], and MODEL POISONING [22, 36, 123]. Targeted poisoning tries to violate the integrity of a few targeted samples and assumes that the attacker has training data control to insert the poisoned samples. Backdoor poisoning attacks require the generation of a BACKDOOR PATTERN, which is added to both the poisoned samples and the testing samples to cause misclassification. Backdoor attacks are the only attacks in the literature that require both training and testing data control. Model poisoning attacks could result in either targeted or backdoor attacks, and the attacker modifies model parameters to cause an integrity violation. They have been designed for centralized learning [222] and federated learning [22, 36].

Privacy compromise [NISTAML.03] [Back to Index]. A PRIVACY COMPROMISE attack causes the unintended leakage of restricted or proprietary information from a PredAI system, including details about a model's training data, weights, or architecture [100, 309]. While the term "confidentiality" is more widely used in taxonomies of traditional cybersecurity attacks, the AML field has tended to use the top-level term "privacy" to encompass both attacks against the confidentiality of a model (e.g., those that extract information about a model's weights or architecture) and those that cause violations of expected privacy properties of model outputs (e.g. by exposing model training data) [310]. DATA CONFIDENTIALITY during ML training can be achieved through secure computation methods based on cryptographic techniques [2, 253, 288, 385], which ensure that training data and model parameters remain protected during the training phase. However, even models trained using paradigms that enforce data confidentiality may be vulnerable to privacy attacks, in which adversaries interacting with a model can extract information about its training data or parameters. In this report, we focus on privacy compromises that can occur at deployment time, regardless of the training method used, or whether data confidentiality was maintained during training.

In privacy attacks, attackers might be interested in learning information about the training data (resulting in DATA PRIVACY ATTACKS) or the ML model (resulting in MODEL PRIVACY ATTACKS). The attacker could have different objectives for compromising the privacy of training data, such as DATA RECONSTRUCTION [110] (inferring the content or features of training data), MEMBERSHIP-INFERENCE ATTACK [162, 343] (inferring the presence of data in the training set), TRAINING DATA EXTRACTION [59, 63] (extracting training data from generative models), ATTRIBUTE INFERENCE ATTACKS [184, 409] (inferring sensitive attributes of training records) and PROPERTY INFERENCE [134] (inferring properties about the training data distribution). MODEL EXTRACTION is a model privacy attack in which attackers aim to extract information about the model [177].

2.1.3. Attacker Capabilities

AML attacks for PredAI systems can be taxonomized with respect to the capabilities that an attacker controls. An adversary might leverage six types of capabilities to achieve their objectives, as shown in the outer layer of the objective circles in Fig. 1:

- TRAINING DATA CONTROL: The attacker might take control of a subset of the training data by inserting or modifying training samples. This capability is used in data poisoning attacks (e.g., availability poisoning, targeted or backdoor poisoning).

- MODEL CONTROL: The attacker might take control of the model parameters by either 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 might add perturbations to testing samples at model deployment time, as performed in evasion attacks to generate adversarial examples or in backdoor poisoning attacks.
- LABEL LIMIT: This capability is relevant to restrict adversarial control over the labels of training samples in supervised learning. Clean-label poisoning attacks assume that the attacker does not control the label of the poisoned samples, while regular poisoning attacks assume label control over the poisoned samples.
- SOURCE CODE CONTROL: The attacker might modify the source code of the ML algorithm, such as the random number generator or any third-party libraries, which are often open source.
- QUERY ACCESS: The attacker might submit queries to the model and receive predictions (i.e., labels or model confidences), such as when interacting with an AI system hosted by a cloud provider as a machine learning as a service (MLaaS) offering. This capability is used by black-box evasion attacks, ENERGY-LATENCY ATTACK, and all privacy attacks that do not require knowledge of the model's training data, architecture, or parameters.

Even if an attacker does not have the ability to modify training/testing data, source code, or model parameters, access to these may still be crucial for mounting stronger white-box attacks that require knowledge of the ML system. See Sec. 2.1.4 for more details on attacker knowledge, and detailed definitions of white-box and black-box attacks.

Figure 1 connects each attack class with the capabilities required to mount the attack. For example, backdoor attacks that cause integrity violations require control of the training and testing data to insert the backdoor pattern. Backdoor attacks can also be mounted via source code control, particularly when training is outsourced to a more powerful entity. Clean-label backdoor attacks do not allow label control on the poisoned samples in addition to the capabilities needed for backdoor attacks.

2.1.4. Attacker Knowledge

Another dimension of attack classification is how much knowledge the attacker has about the ML system. There are three main types of attacks:

White-box attacks. These assume that the attacker operates with *full* knowledge about the ML system, including the training data, model architecture, and model hyperparameters. While these attacks operate under very strong assumptions, the main reason for analyzing them is to test the vulnerability of a system against worst-case adversaries and to evaluate potential mitigations. This definition is more general and encompasses the notion of adaptive attacks in which knowledge of the mitigations applied to the model or the system is explicitly tracked.

Black-box attacks. These attacks assume that the attacker operates with minimal, and sometimes no knowledge at all about the ML system. An adversary might have query access to the model, but they have no other information about how the model is trained. These attacks are the most practical since they assume that the attacker has no knowledge of the AI system and utilizes system interfaces readily available for normal use.

Gray-box attacks. There are a range of gray-box attacks that capture adversarial knowledge between black-box and white-box attacks. Suciu et al. [358] introduced a framework to classify gray-box attacks. An attacker might know the model architecture but not its parameters, or the attacker might know the model and its parameters but not the training data. Other common assumptions for gray-box attacks are that the attacker has access to data distributed identically to the training data and knows the feature representation. The latter assumption is important for applications in which feature extraction is used before training an ML model, such as cybersecurity, finance, and healthcare.

2.1.5. Data Modality

Until recently, most attacks and defenses in adversarial machine learning have operated under a single modality, but a new trend in the field is to use multimodal data. The taxonomy of attacks defined in Fig. 1 is independent of the modality of the data in specific applications.

The most common data modalities in the AML literature include:

- **Image:** Adversarial examples of image data [144, 362] have the advantage of a continuous domain, and gradient-based methods can be applied directly for optimization. Backdoor poisoning attacks were first invented for images [148], and many privacy attacks are run on image datasets (e.g., [342]). The image modality includes other types of imaging (e.g., LIDAR, SAR, IR, hyperspectral).
- **Text:** Text is a popular modality, and all classes of attacks have been proposed for text models, including evasion [150], poisoning [82, 213], and privacy [426].
- **Audio:** Audio systems and text generated from audio signals have also been attacked [66].
- **Video:** Video comprehension models have shown increasing capabilities in vision and language tasks [428], but such models are also vulnerable to attacks [402].
- **Cybersecurity**[2]**:** The first poisoning attacks were discovered in cybersecurity for worm

[2] Cybersecurity data may not include a single modality but rather multiple modalities, such as network-level, host-level, or program-level data.

signature generation (2006) [291] and spam email classification (2008) [269]. Since then, poisoning attacks have been shown for malware classification, malicious PDF detection, and Android malicious app classification [329]. Evasion attacks against similar data modalities have been proposed as well: malware classification [103, 357], PDF malware classification [352, 414], Android malicious app detection [295], and network instrusion detection [93]. Poisoning unsupervised learning models has been shown for clustering used in malware classification [41] and network traffic anomaly detection [315].

Anomaly detection based on data-centric approaches allows for automated feature learning through ML algorithms. However, the application of ML to such problems comes with specific challenges related to the need for very low false negative and low false positive rates (e.g., the ability to catch zero-day attacks). This challenge is compounded by the fact that trying to accommodate all of these together makes ML models susceptible to adversarial attacks [198, 301, 446].

- **Tabular data:** There have been numerous attacks against ML models working on tabular data, such as poisoning availability attacks against healthcare and business applications [179], privacy attacks against healthcare data [422], and evasion attacks against financial applications [141].

Recently, the use of ML models trained on multimodal data has gained traction, particularly the combination of image and text data modalities. Several papers have shown that multimodal models may provide some resilience against attacks [417], but other papers show that multimodal models themselves could be vulnerable to attacks mounted on all modalities at the same time [77, 333, 415] (see Sec. 4.2.3).

> An open challenge is to test and characterize the resilience of a variety of multimodal ML models against evasion, poisoning, and privacy attacks.

2.2. Evasion Attacks and Mitigations

[**NISTAML.022**] [Back to Index]

The discovery of evasion attacks against ML models has led to significant growth in AML research over the last decade. In an evasion attack, the adversary's goal is to generate adversarial examples: samples whose classification can be changed to an arbitrary class of the attacker's choice – often with only minimal perturbation [362]. For example, in the context of image classification, the perturbation of the original sample might be small so that a human cannot observe the transformation of the input; while the ML model can be tricked to classify the adversarial example in the target class selected by the attacker, humans still recognize it as part of the original class.

Early known instances of evasion attacks date back to 1988 with the work of Kearns and Li [192] and 2004 when Dalvi et al. [98] and Lowd and Meek [226] demonstrated the existence of adversarial examples for linear classifiers used in spam filters. Later, Szedegy et al. [362] showed that deep neural networks used for image classification could be easily manipulated through adversarial examples. In 2013, Szedegy et al. [362] and Biggio et al. [38] independently discovered an effective method for generating adversarial examples against linear models and neural networks by applying gradient optimization to an adversarial objective function. Both of these techniques require white-box access to the model and were improved by subsequent methods that generated adversarial examples with even smaller perturbations [20, 65, 232].

Adversarial examples are also applicable in more realistic black-box settings in which attackers only obtain query access capabilities to the trained model. Even in the more challenging black-box setting in which attackers obtain the model's predicted labels or confidence scores, deep neural networks are still vulnerable to adversarial examples. Methods for creating adversarial examples in black-box settings include zeroth-order optimization [80], discrete optimization [254], Bayesian optimization [344], and *transferability*, which involves the white-box generation of adversarial examples on a different model before transferring them to the target model [282, 283, 377]. While cybersecurity and image classifications were the first application domains to showcase evasion attacks, ML technology in many other application domains has come under scrutiny, including speech recognition [66], natural language processing [185], and video classification [215, 401].

Mitigating adversarial examples is a well-known challenge in the community and deserves additional research and investigation. The field has a history of publishing defenses evaluated under relatively weak adversarial models that are subsequently broken by more powerful attacks. Mitigations need to be evaluated against strong adaptive attacks, and guidelines for the rigorous evaluation of newly proposed mitigation techniques have been established [97, 375]. The most promising directions for mitigating the critical threat of evasion attacks are adversarial training [144, 232] (iteratively generating and inserting adversarial examples with their correct labels at training time); certified techniques, such as randomized smoothing [94] (evaluating ML prediction under noise); and formal verification techniques [136, 191] (applying formal method techniques to verify the model's output). Nevertheless, these methods have different limitations, such as decreased accuracy for adversarial training and randomized smoothing and computational complexity for formal methods. There is an inherent trade-off between robustness and accuracy [374, 379, 433]. Similarly, there are trade-offs between a model's robustness and fairness guarantees [71].

2.2.1. White-Box Evasion Attacks

In the white-box threat model, the attacker has full knowledge of the model architecture and parameters, as discussed in Section 2.1.4. The main challenge for creating adversarial examples in this setting is to find a perturbation added to a testing sample that changes its classification label, often with constraints on properties such as the perceptibility or size of the perturbation. In the white-box threat model, it is common to craft adversarial examples by solving an optimization problem written from the attacker's perspective, which specifies the objective function for the optimization (such as changing the target label to a certain class), as well as a distance metric to measure the similarity between the testing sample and the adversarial example.

Optimization-based methods. Szedegy et al. [362] and Biggio et al. [38] independently proposed the use of optimization techniques to generate adversarial examples. In their threat models, the adversary is allowed to inspect the entirety of the ML model and compute gradients relative to the model's loss function. These attacks can be targeted (i.e., the adversarial example's class is selected by the attacker) or untargeted (i.e., the adversarial examples are misclassified to any other incorrect class).

Szedegy et al. [362] coined the widely used term *adversarial examples*. They considered an objective that minimized the ℓ_2 norm of the perturbation subject to the model prediction changing to the target class. The optimization is solved using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method. Biggio et al. [38] considered the setting of a binary classifier with malicious and benign classes with a continuous and differentiable discriminant function. The objective of the optimization is to minimize the discriminant function in order to generate adversarial examples of maximum confidence.

While Biggio et al. [38] applied their method to linear classifiers, kernel SVM, and multi-layer perceptrons, Szedegy et al. [362] showed the existence of adversarial examples on deep learning models used for image classification. Goodfellow et al. [144] introduced an efficient method for generating adversarial examples for deep learning: the Fast Gradient Sign Method (FGSM), which performs a single iteration of gradient descent for solving the optimization. This method has been extended to an iterative FGSM attack by Kurakin et al. [200].

Subsequent works have proposed new objectives and methods for optimizing the generation of adversarial examples with the goals of minimizing the perturbations and supporting multiple distance metrics. Some notable attacks include:

- DeepFool is an untargeted evasion attack for ℓ_2 norms, which uses a linear approximation of the neural network to construct the adversarial examples [257].
- The Carlini-Wagner attack uses multiple objectives that minimize the loss or logits on the target class and the distance between the adversarial example and original sample. The attack is optimized via the penalty method [65] and considers three distance metrics to measure the perturbations of adversarial examples: ℓ_0, ℓ_2, and ℓ_∞. The attack has been effective against the defensive distillation defense [284].
- The Projected Gradient Descent (PGD) attack [232] minimizes the loss function and projects the adversarial examples to the space of allowed perturbations at each iteration of gradient descent. PGD can be applied to the ℓ_2 and ℓ_∞ distance metrics for measuring the perturbation of adversarial examples.

Universal evasion attacks. Moosavi-Dezfooli et al. [256] showed how to construct small universal perturbations (with respect to some norm) that can be added to most images and induce a misclassification. Their technique relies on successive optimization of the universal perturbation using a set of points sampled from the data distribution. This is a form of FUNCTIONAL ATTACK. An interesting observation is that the universal perturbations generalize across deep network architectures, suggesting similarity in the decision boundaries trained by different models for the same task.

Physically realizable attacks. These are attacks against ML systems that can be implemented feasibly in the physical world [21, 200, 227]. One of the first instances was the attack on facial recognition systems by Sharif et al. [332]. The attack can be realized by printing a pair of eyeglass frames, which misleads facial recognition systems to either evade detection or impersonate another individual. Eykholt et al. [122] proposed an attack to generate robust perturbations under different conditions, resulting in adversarial examples that can evade vision classifiers in various physical environments. The attack is applied to evade a road sign detection classifier by physically applying black and white stickers to the road signs. The ShapeShifter [81] attack was designed to evade object detectors, which is a more challenging problem than attacking image classifiers since the attacker needs to evade the classification in multiple bounding boxes with different scales. This attack also requires the perturbation to be robust enough to survive real-world distortions due to different viewing distances, angles, lighting conditions, and camera limitations.

Other data modalities. In computer vision applications, adversarial examples are often designed to be imperceptible to humans. Therefore, the perturbations introduced by attackers need to be so small that a human correctly recognizes the images, while the ML classifier is tricked into changing its prediction. Alternatively, there may be a trigger object in the image that is still imperceptible or innocuous to humans but causes the model to misclassify. The concept of adversarial examples has been extended to other domains, such as audio, video, NLP, and cybersecurity. In some of these settings, there are additional constraints that need to be respected by adversarial examples, such as text semantics in NLP and the application constraints in cybersecurity. Several representative works include:

- **Audio:** Carlini and Wagner [66] showed a targeted attack on models that generate text from speech. They can generate an audio waveform that is very similar to an existing one but that can be transcribed to any text of the attacker's choice.
- **Video:** Adversarial evasion attacks against video classification models can be split into sparse attacks that perturb a small number of video frames [401] and dense attacks that perturb all of the frames in a video [215]. The goal of the attacker is to change the classification label of the video.
- **Text:** Jia and Liang [185] developed a methodology for generating adversarial text examples. This pioneering work was followed by many advances in developing adversarial attacks on natural language processing (NLP) models (see a comprehensive survey on the topic [438]). La Malfa and Kwiatkowska [202] proposed a method for formalizing perturbation definitions in NLP by introducing the concept of semantic robustness. The main challenges in NLP are that the domain is discrete rather than continuous (e.g., image, audio, and video classification), and adversarial examples need to respect text semantics. These challenges are illustrated by the recent ASCII-art attack [186] against chatbots. An ASCII-art illustration of a forbidden term tricks the chatbot into providing the harmful information even when the chatbot correctly censors the plain English word. The semantic distance between the two prompts is precisely zero, and both of them should have been treated the same.
- **Cybersecurity:** In cybersecurity applications, adversarial examples must respect the constraints imposed by the application semantics and feature representation of cyber data, such as network traffic or program binaries. FENCE is a general framework for crafting white-box evasion attacks using gradient optimization in discrete domains and supports a range of linear and statistical feature dependencies [88]. FENCE has been applied to two network security applications: malicious domain detection and malicious network traffic classification. Sheatsley et al. [334] proposed a method that learns the constraints in feature space using formal logic and crafts adversarial examples by projecting them onto a constraint-compliant space. They applied the technique to network intrusion detection and phishing classifiers. Both papers observed that attacks from continuous domains cannot be readily applied in constrained environments, as they result in infeasible adversarial examples. Pierazzi et al. [295] discussed the difficulty of mounting feasible evasion attacks in cybersecurity due to constraints in feature space and the challenge of mapping attacks from feature space to problem space. They formalized evasion attacks in problem space and constructed feasible adversarial examples for Android malware.

2.2.2. Black-Box Evasion Attacks

[**NISTAML.025**] [Back to Index]

Black-box evasion attacks are designed under a realistic adversarial model in which the attacker has no prior knowledge of the model architecture or training data. Instead, the adversary can interact with a trained ML model by querying it on various data samples and obtaining the model's predictions. Similar APIs are provided by MLaaS offered by public cloud providers, in which users can obtain the model's predictions on selected queries without information about how the model was trained. There are two main classes of black-box evasion attacks in the literature:

- **Score-based attacks:** In this setting, attackers obtain the model's confidence scores or logits and can use various optimization techniques to create the adversarial examples. A popular method is zeroth-order optimization, which estimates the model's gradients without explicitly computing derivatives [80, 173]. Other optimization techniques include discrete optimization [254], natural evolution strategies [172], and random walks [262].
- **Decision-based attacks:** In this more restrictive setting, attackers only obtain the final predicted labels of the model. The first method for generating evasion attacks was the Boundary Attack based on random walks along the decision boundary and rejection sampling [47], which was extended with an improved gradient estimation to reduce the number of queries in the HopSkipJumpAttack [79]. More recently, several optimization methods search for the direction of the nearest decision boundary (e.g., the OPT attack [86]), use sign SGD instead of binary searches (e.g., the Sign-OPT attack [87]), or use Bayesian optimization [344].

> The primary challenge in creating adversarial examples in black-box settings is reducing the number of queries to the ML models. Recent techniques can successfully evade the ML classifiers with a relatively small number of queries, typically less than 1000 [344].

2.2.3. Transferability of Attacks

Another method for generating adversarial attacks under restrictive threat models involves transferring an attack crafted on a different ML model. Typically, an attacker trains a substitute ML model, generates white-box adversarial attacks on the substitute model, and transfers the attacks to the target model. Various methods differ in how the substitute models are trained. For example, Papernot et al. [282, 283] train the substitute model with score-based queries to the target model, while several papers train an ensemble of models without explicitly querying the target model [218, 377, 397].

Attack transferability is an intriguing phenomenon, and existing literature attempts to understand the fundamental reasons why adversarial examples transfer across models. Several papers have observed that different models learn intersecting decision boundaries in both benign and adversarial dimensions, which leads to better transferability [144, 256, 377]. Demontis et al. [104] identified two main factors that contribute to attack transferability for both evasion and poisoning: the intrinsic adversarial vulnerability of the target model and the complexity of the surrogate model used to optimize the attack. EXPECTATION OVER TRANSFORMATION aims to make adversarial examples sustain image transformations that occur in the real world, such as angle and viewpoint changes [21].

2.2.4. Evasion attacks in the real world

While many of the attacks discussed in this section were demonstrated only in research settings, several evasion attacks have been demonstrated in the real world, and we discuss prominent instances in face recognition systems, phishing webpage detection, and malware classification.

Face recognition systems used for identity verification have been the target of adversarial evasion attacks, as they constitute an entry point to critical systems and enable users to commit financial fraud. During the last half of 2020, the ID.me face recognition service found more than 80,000 attempts of users attempting to fool their ID verification steps used by multiple state workforce agencies [276]. These attacks included people wearing masks, using deepfakes, or using images or videos of other people. The intent was to fraudulently claim unemployment benefits provided during COVID relief efforts. Later in 2022, according to US federal prosecutors, a New Jersey man was able to verify fake driver's licenses through ID.me as part of a US$ 2.5M unemployment-fraud scheme. This time, the suspect used various wigs to evade the face recognition system [156].

Another case study of real-world evasion attacks reported by Apruzzese et al. [17] is an attack against a commercial phishing webpage detector. The ML phishing detector is an ensemble of multiple models that analyze different aspects of the image to determine if it is a phishing attempt. Inputs that are marked uncertain by the model are triaged to security analysts. Out of 4600 samples marked uncertain by the ML image classification system, the authors identified 100 adversarial examples. Interestingly, a manual analysis of these adversarial examples revealed that attackers do not employ optimization-based attacks, but rather utilize relatively simple methods for evasion, such as image cropping, masking, or blurring techniques.

Other examples of evasion attacks demonstrated by researchers in malware classification are cataloged in the MITRE Adversarial Threat Landscape for Artificial-Intelligence Systems (ATLAS) knowledge base [248]. Palo Alto Networks reported evasion attacks against a deep learning detector for malware command-and-control traffic, and a botnet Domain Generation Algorithm (DGA) detector. An instance of a universal evasion attack was discovered against Cylance's AI malware detection model. Researchers also evaded ProofPoint's email protection system by training a shadow ML model and using the insights from that to attack the real system. These are demonstrations of evasion vulnerabilities by researchers, but did not result in attacks in the wild.

2.2.5. Mitigations

Mitigating evasion attacks is challenging because adversarial examples are widespread in a variety of ML model architectures and application domains. Possible explanations for the existence of adversarial examples are that ML models rely on non-robust features that are not aligned with human perception in the computer vision domain [174]. In the last few years, many of the proposed mitigations against adversarial examples have been ineffective against stronger attacks. Furthermore, several papers have performed extensive evaluations and defeated a large number of proposed mitigations:

- Carlini and Wagner showed how to bypass 10 methods for detecting adversarial examples and described several guidelines for evaluating defenses [64]. Recent work shows that detecting adversarial examples is as difficult as building a defense [373]. Therefore, this direction for mitigating adversarial examples is similarly challenging as designing defenses.
- The Obfuscated Gradients attack [20] was specifically designed to defeat several proposed defenses that rely on masking gradients to protect against optimization-based attacks. It relies on a new technique, Backward Pass Differentiable Approximation, which approximates the gradient during the backward pass of backpropagation, and was shown to bypass several proposed defenses based on gradient masking.
- Tramèr et al. [375] described a methodology for designing adaptive attacks against proposed defenses and circumvented 13 existing defenses. They advocate for designing adaptive attacks to test newly proposed defenses rather than merely testing the defenses against well-known attacks.

From the wide range of proposed defenses against adversarial evasion attacks, three main classes have proven to be resilient and have the potential to provide mitigation against evasion attacks:

1. **Adversarial training:** Introduced by Goodfellow et al. [144] and further developed by Madry et al. [232], adversarial training is a general method that augments training data with adversarial examples generated iteratively during training using their correct labels. The stronger the adversarial attacks for generating adversarial examples are, the more resilient the trained model becomes. Adversarial training results in models with more semantic meaning than standard models [379], but this benefit usually comes at the cost of decreased model accuracy on clean data. Additionally, adversarial training is expensive due to the iterative generation of adversarial examples during training.
2. **Randomized smoothing:** Proposed by Lecuyer et al. [207] and further improved by

Cohen et al. [94], randomized smoothing is a method that transforms any classifier into a certifiable robust smooth classifier by producing the most likely predictions under Gaussian noise perturbations. This method results in provable robustness for ℓ_2 evasion attacks, even for classifiers trained on large-scale datasets, such as ImageNet. Randomized smoothing typically provides certified prediction to a subset of testing samples, the exact number of which depends on factors such as the size of the potential perturbations or the characteristics of the training data and model. Recent results have extended the notion of certified adversarial robustness to ℓ_2-norm bounded perturbations by combining a pretrained denoising diffusion probabilistic model and a standard high-accuracy classifier [62]. Li et al. [211] developed a taxonomy for the robustness verification and training of representative algorithms. They also revealed the characteristics, strengths, limitations, and fundamental connections among these approaches, along with theoretical barriers facing the field.

3. **Formal verification:** Another method for certifying the adversarial robustness of a neural network is based on techniques from FORMAL METHODS. Reluplex uses satisfiability modulo theories (SMT) solvers to verify the robustness of small feed-forward neural networks [191]. AI^2 is the first verification method applicable to convolutional neural networks using abstract interpretation techniques [136]. These methods have been extended and scaled up to larger networks in follow-up verification systems, such as DeepPoly [346], ReluVal [394], and Fast Geometric Projections (FGP) [131]. Formal verification techniques have significant potential for certifying neural network robustness but are limited by their lack of scalability, computational cost, and restriction in the type of supported algebraic operations such as addition, multiplication, etc.

All of these proposed mitigations exhibit inherent trade-offs between robustness and accuracy, and they come with additional computational costs during training. Therefore, designing ML models that resist evasion while maintaining accuracy remains an open problem. See Section 4.1.1 for further discussion on these trade-offs.

2.3. Poisoning Attacks and Mitigations

Poisoning attacks are broadly defined as adversarial attacks during the training stage of the ML algorithm. The first known poisoning attack was developed for worm signature generation in 2006 [291]. Since then, poisoning attacks have been studied extensively in several application domains: computer security (for spam detection [269], network intrusion detection [384], vulnerability prediction [318], malware classification [329, 412]), computer vision [137, 148, 330], natural language processing (NLP) [82, 213, 388], and tabular data in healthcare and financial domains [179]. Recently, poisoning attacks have gained more attention in industry applications as well [199]. They can even be orchestrated at scale so that an adversary with limited financial resources could control a fraction of the public datasets used for model training [57].

Poisoning attacks are powerful and can cause availability or integrity violations. Availability poisoning attacks typically cause indiscriminate degradation of the ML model on all samples, while targeted and backdoor poisoning attacks induce integrity violations on a small set of target samples. Poisoning attacks leverage a wide range of adversarial capabilities (e.g., data poisoning, model poisoning, label control, source code control, and test data control), resulting in several subcategories of poisoning attacks. They have been developed in white-box [40, 179, 412], gray-box [179], and black-box settings [39].

This section describes availability poisoning, targeted poisoning, backdoor poisoning, and model poisoning attacks classified according to their adversarial objective. For each poisoning attack category, techniques for mounting the attacks, existing mitigations, and their limitations are also discussed. The classification of poisoning attacks in this document is inspired by the framework developed by Cinà et al. [91], which includes additional references to poisoning attacks and mitigations.

2.3.1. Availability Poisoning

[**NISTAML.013**] [Back to Index]

The first poisoning attacks discovered in cybersecurity applications were availability attacks against worm signature generation and spam classifiers, which indiscriminately degrade the performance of the entire ML model in order to effectively prevent its use. Perdisci et al. [291] generated suspicious flows with fake invariants that mislead the worm signature generation algorithm in Polygraph [270]. Nelson et al. [269] designed poisoning attacks against Bayes-based spam classifiers by generating training samples of ”spam” emails containing long sequences of words that appear in legitimate emails, degrading the performance of the spam classifier by inducing a higher rate of false positives. Both of these attacks were conducted under the white-box setting in which adversaries were aware of the ML training algorithm, feature representations, training datasets, and ML models. Availability poisoning attacks have also been proposed for ML-based systems that detect cybersecurity attacks against industrial control systems: such detectors are often retrained using data collected during system operation to account for plant operational drift of the monitored signals, creating opportunities for an attacker to mimic the signals of corrupted sensors at training time to poison the detector such that real attacks remain undetected at deployment time [198].

A simple black-box poisoning attack strategy is LABEL FLIPPING, in which an adversary generates training examples with incorrect or altered labels [39]. This method may require a large percentage of poisoning samples to mount an availability attack. These attacks can also be formulated through optimization-based methods, such as by solving a bilevel optimization problem to determine the optimal poisoning samples that will achieve the adversarial objective (i.e., maximize the hinge loss for a SVM [40] or maximize the mean square error [MSE] for regression [179]). Similar optimization-based availability poisoning attacks have been designed against linear regression [179] and neural networks [260], although these optimization-based attacks may require white-box access to the model and training data. In gray-box adversarial settings, the most popular method for generating availability poisoning attacks is transferability, in which poisoning samples are generated for a surrogate model and transferred to the target model [104, 358].

Clean-label poisoning [NISTAML.012] [Back to Index]. A realistic threat model for supervised learning is that of clean-label poisoning attacks, in which adversaries can only control the training examples but not their labels. This may arise in scenarios in which the labeling process is external to the training algorithm, as in malware classification where binary files can be submitted by attackers to threat intelligence platforms and labeling is performed using anti-virus signatures or other external methods. Clean-label availability attacks have been introduced for neural network classifiers by training a generative model and adding noise to training samples to maximize the adversarial objective [128]. A different approach for clean-label poisoning is to use gradient alignment and minimally modify the training data [129].

Availability poisoning attacks have also been designed for unsupervised learning against centroid-based anomaly detection [195] and behavioral clustering for malware [41]. In federated learning, an adversary can mount a model poisoning attack to induce availability violations in the globally trained model [123, 335, 336]. More details on model poisoning attacks are provided in Sec. 2.3.

Mitigations. Availability poisoning attacks are usually detectable by monitoring the standard performance metrics of ML models (e.g., precision, recall, accuracy, F1 scores, and area under the curve) as they cause a large degradation in the classifier metrics. However, detecting these attacks during the testing or deployment stages of ML may be less desirable, and many existing mitigations aim to proactively prevent these attacks during the training stage to generate robust ML models. Existing mitigations for availability poisoning attacks include:

- **Training data sanitization:** These methods leverage the insight that poisoned samples are typically different than regular training samples that are not controlled by adversaries. As such, data sanitization techniques are designed to clean the training set and remove the poisoned samples before the ML training is performed. Cretu et al. [96] proposed the first sanitization procedure for unlabeled datasets that relies on majority voting of multiple models trained on subsets of the training set. They apply the method to anomaly detection on network packets. Nelson et al. [269] introduced the Region of Non-Interest (RONI) method, which examines each sample and excludes it from training if the accuracy of the model decreases when the sample is added. Subsequently proposed sanitization methods improved upon these early approaches by reducing their computational complexity and considering other applications. Paudice et al. [289] introduced a method for label cleaning that was specifically designed for label-flipping attacks. Steinhardt et al. [354] proposed the use of outlier detection methods for identifying poisoned samples. Clustering methods have also been used to detect poisoned samples [203, 363]. Other work has suggested that computing the variance of predictions made by an ensemble of multiple ML models is an effective data sanitization method for network intrusion detection [384]. Once sanitized, datasets may be protected by cybersecurity mechanisms for provenance and integrity attestation [267].

- **Robust training:** An alternative approach to mitigating availability poisoning attacks is to modify the ML training algorithm to increase the robustness of the resulting model. The defender can train an ensemble of multiple models and generate predictions via model voting [37, 209, 395]. Several papers apply techniques from robust optimization, such as using a trimmed loss function [109, 179]. Rosenfeld et al. [314] proposed the use of randomized smoothing to add noise during training to provide protection against label-flipping attacks.

2.3.2. Targeted Poisoning

[**NISTAML.024**] [Back to Index]

In contrast to availability attacks, targeted poisoning attacks induce a change in the ML model's prediction on a small number of targeted samples. If the adversary can control the labeling function of the training data, then label-flipping is an effective targeted poisoning attack: the adversary simply inserts several poisoned samples with the target label, and the model will learn the wrong label. Therefore, targeted poisoning attacks are mostly studied in a clean-label setting in which the attacker does not have control over training data labels.

Several techniques for mounting clean-label targeted attacks have been proposed. Koh and Liang [196] showed how influence functions (i.e., a statistical method that determines the most influential training samples for a prediction) can be leveraged to create poisoned samples in the fine-tuning setting in which a pre-trained model is fine-tuned on new data. Suciu et al. [358] designed StingRay, a targeted poisoning attack that modifies samples in feature space and adds poisoned samples to each mini batch of training. An optimization procedure based on feature collision was crafted by Shafahi et al. [330] to generate clean-label targeted poisoning for fine-tuning and end-to-end learning. ConvexPolytope [444] and BullseyePolytope [4] optimized the poisoning samples against ensemble models, which offers better advantages for attack transferability. MetaPoison [166] uses a meta-learning algorithm to optimize the poisoned samples, while Witches' Brew [137] performs optimization by gradient alignment, resulting in a state-of-the-art targeted poisoning attack.

All of the above attacks impact a small set of targeted samples that are selected by the attacker during training, and they have only been tested for continuous image datasets (with the exception of StingRay, which requires adversarial control of a large fraction of the training set). Subpopulation poisoning attacks [180] were designed to poison samples from an entire subpopulation, defined by matching on a subset of features or creating clusters in representation space. Poisoned samples are generated using label-flipping (for NLP and tabular modalities) or a first-order optimization method (for continuous data, such as images). The attack generalizes to all samples in a subpopulation and requires minimal knowledge about the ML model and a small number of poisoned samples proportional to the subpopulation size.

Targeted poisoning attacks have also been introduced for semi-supervised learning algorithms [53], such as MixMatch [34], FixMatch [347], and Unsupervised Data Augmentation (UDA) [413] in which the adversary poisons a small fraction of the unlabeled training dataset to change the prediction on targeted samples at deployment time.

Mitigations. Targeted poisoning attacks are notoriously challenging to defend against. Jagielski et al. [180] showed an impossibility result for subpopulation poisoning attacks. To mitigate some of the risks associated with such attacks, model developers may protect training data through traditional cybersecurity measures such as access controls, use methods for data sanitization and validation, and use mechanisms for dataset provenance and integrity attestation [267]. Ma et al. [230] proposed the use of differential privacy (DP) as a defense (which follows directly from the definition of differential privacy), but differentially private ML models may also have lower accuracy than standard models, and the trade-off between robustness and accuracy needs to be considered in each application. See Section 4.1.1 for further discussion on the trade-offs between the attributes of Trustworthy AI systems.

2.3.3. Backdoor Poisoning

[**NISTAML.021, NISTAML.023**] [Back to Index]

Backdoor poisoning attacks are poisoning attacks that cause the targeted model to misclassify samples containing a particular BACKDOOR PATTERN or trigger. In 2017, Gu et al. [148] proposed BadNets, the first backdoor poisoning attack. They observed that image classifiers can be poisoned by adding a small patch trigger in a subset of images at training time and changing their label to a target class. The classifier learns to associate the trigger with the target class, and any image that includes the trigger or backdoor pattern will be misclassified to the target class at testing time. Concurrently, Chen et al. [84] introduced backdoor attacks in which the trigger is blended into the training data. Follow-up work introduced the concept of clean-label backdoor attacks [380] in which the adversary cannot change the label of the poisoned examples. Clean-label attacks typically require more poisoning samples to be effective, but the attack model is more realistic.

In the last few years, backdoor attacks have become more sophisticated and stealthy, making them harder to detect and mitigate. Latent backdoor attacks were designed to survive even upon model fine-tuning of the last few layers using clean data [420]. Backdoor Generating Network (BaN) [322] is a dynamic backdoor attack in which the location of the trigger changes in the poisoned samples so that the model learns the trigger in a location-invariant manner. Functional triggers (i.e., FUNCTIONAL ATTACK) are embedded throughout the image or change according to the input. Li et al. used steganography algorithms to hide the trigger in the training data [214] and introduced a clean-label attack that uses natural reflection on images as a backdoor trigger [223]. Wenger et al. [404] poisoned facial recognition systems by using physical objects as triggers, such as sunglasses and earrings. Architectural backdoor attacks [205] perform malicious modifications to the structure of an ML model during its training phase, which allows an attacker to manipulate the model's behavior when presented with specific triggers. These attacks require adversarial access to the model design or training environment and are applicable when model training is outsourced to a more powerful entity, such as a cloud service.

Other data modalities. While the majority of backdoor poisoning attacks are designed for computer vision applications, this attack vector has been effective in other application domains with different data modalities, such as audio, NLP, and cybersecurity settings.

- **Audio:** In audio domains, Shi et al. [341] showed how an adversary can inject an unnoticeable audio trigger into live speech, which is jointly optimized with the target model during training.
- **NLP:** In NLP, the construction of meaningful poisoning samples is more challenging as the text data is discrete, and the semantic meaning of sentences would ideally be preserved for the attack to remain unnoticeable. Recent work has shown that backdoor attacks in NLP domains are becoming feasible. For instance, Chen et al. [82] introduced semantic-preserving backdoors at the character, word, and sentence level for sentiment analysis and neural machine translation applications. Li et al. [213] generated hidden backdoors against transformer models using generative language models in three NLP tasks: toxic comment detection, neural machine translation, and question answering.
- **Cybersecurity:** Following early work on poisoning in cybersecurity [269, 291], Severi et al. [329] showed how AI explainability techniques can be leveraged to generate clean-label poisoning attacks with small triggers against malware classifiers. They attacked multiple models (i.e., neural networks, gradient boosting, random forests,

and SVMs) using three malware datasets: Ember for Windows PE file classification, Contagio for PDF file classification, and DREBIN for Android app classification. Jigsaw Puzzle [418] designed a backdoor poisoning attack for Android malware classifiers that uses realizable software triggers harvested from benign code.

Mitigations. The literature on backdoor attack mitigation is vast compared to other poisoning attacks. Below we discuss several classes of defenses, including data sanitization, trigger reconstruction, and model inspection and sanitization, and we also mention their limitations.

- **Training data sanitization:** Similar to poisoning availability attacks, training data sanitization can be applied to detecting backdoor poisoning attacks. For example, outlier detection in the latent feature space [157, 293, 378] has been effective for convolutional neural networks used for computer vision applications. Activation Clustering [76] clusters training data in representation space to isolate the backdoored samples in a separate cluster. Data sanitization achieves better results when the poisoning attack controls a relatively large fraction of training data but is not as effective against stealthy poisoning attacks. Overall, this leads to a trade-off between attack success and the detectability of malicious samples.
- **Trigger reconstruction:** This class of mitigations aims to reconstruct the backdoor trigger, assuming that it is at a fixed location in the poisoned training samples. NeuralCleanse by Wang et al. [390] developed the first trigger reconstruction approach and used optimization to determine the most likely backdoor pattern that reliably misclassifies the test samples. The initial technique has been improved to reduce performance time on several classes and simultaneously support multiple triggers inserted into the model [163, 411]. A representative system in this class is Artificial Brain Simulation (ABS) by Liu et al. [221], which stimulates multiple neurons and measures the activations to reconstruct the trigger patterns. Khaddaj et al. [193] developed a new primitive for detecting backdoor attacks and a corresponding effective detection algorithm with theoretical guarantees.
- **Model inspection and sanitization:** Model inspection analyzes the trained ML model before its deployment to determine whether it was poisoned. An early work in this space is NeuronInspect [168], which is based on explainability methods to determine different features between clean and backdoored models that are subsequently used for outlier detection. DeepInspect [78] uses a conditional generative model to learn the probability distribution of trigger patterns and performs model patching to remove the trigger. Xu et al. [416] proposed the Meta Neural Trojan Detection (MNTD) framework, which trains a meta-classifier to predict whether a given ML model is backdoored (or "Trojaned," in the authors' terminology). This technique is general and can be applied to multiple data modalities, such as vision, speech, tabular data, and NLP. Once a backdoor is detected, model sanitization can be performed via pruning [407], retraining [429], or fine-tuning [217] to restore the

model's accuracy.

- **Certified defenses:** Several methods for achieving certified defenses against data poisoning attacks have been proposed in the literature. BagFlip [440] is a model-agnostic defense that extends randomized smoothing [94] and combines training data bagging with adding noise to both training and testing samples. Deep Partition Aggregation [209] and Deep Finite Aggregation [396] are certified defenses that partition the training data into disjointed subsets and train an ensemble method on each partition to reduce the impact of poisoned samples. Recently, FCert [398] provides a certified defense against data poisoning in few-shot classification settings used for both vision and text data.

Most of these mitigations have been designed against computer vision classifiers based on convolutional neural networks using backdoors with fixed trigger patterns. Severi et al. [329] showed that some of the data sanitization techniques (e.g., spectral signatures [378] and Activation Clustering [76]) are ineffective against clean-label backdoor poisoning on malware classifiers. More recent semantic and functional backdoor triggers would also pose challenges to approaches based on trigger reconstruction or model inspection, which generally assume fixed backdoor patterns. The limitation of using meta classifiers for predicting a Trojaned model [416] is the high computational complexity of the training stage of the meta classifier, which requires training thousands of SHADOW MODEL. Additional research is required to design strong backdoor mitigation strategies that can protect ML models against this important attack vector without suffering from these limitations.

In cybersecurity, Rubinstein et al. [315] proposed an approach based on principal component analysis (PCA) to mitigate poisoning attacks against PCA subspace anomaly detection methods in backbone networks. It maximized median absolute deviation (MAD) instead of variance to compute principal components and used a threshold value based on Laplace distribution instead of Gaussian. Madani and Vlajic [231] built an autoencoder-based intrusion detection system, assuming that malicious poisoning attack instances were under 2%.

[193] provided a different perspective on backdoor mitigation by showing that backdoors are indistinguishable from naturally occurring features in the data if no additional assumptions are made about the attack. However, assuming that the backdoor creates the strongest feature in the data, the paper proposed an optimization technique to identify and remove the training samples that correspond to the backdoor.

Poison forensics [331] is a technique for root cause analysis that identifies malicious training samples and complements existing mitigations that are not always resilient in the face of evolving attacks. Poison forensics adds another layer of defense in an ML system: once a poisoning attack is detected at deployment time, poison forensics can trace back to the source of the attack in the training set.

2.3.4. Model Poisoning

[**NISTAML.011**, **NISTAML.026**] [Back to Index]

Model poisoning attacks attempt to directly modify the trained ML model to inject malicious functionality into it. In centralized learning, TrojNN [222] reverse engineers the trigger from a trained neural network and then retrains the model by embedding the trigger in external data to poison it. Most model poisoning attacks have been designed in the federated learning setting in which clients send local model updates to a server that aggregates them into a global model. Compromised clients can send malicious updates to poison the global model. Model poisoning attacks can cause both availability and integrity violation in federated models:

- Poisoning availability attacks that degrade the global model's accuracy have been effective, but they usually require a large percentage of clients to be under the control of the adversary [123, 335].
- Targeted model poisoning attacks induce integrity violations on a small set of samples at testing time. They can be mounted by a model replacement or model boosting attack in which the compromised client replaces the local model update according to the targeted objective [23, 35, 360].
- Backdoor model poisoning attacks introduce a trigger via malicious client updates to induce the misclassification of all samples with the trigger at testing time [23, 35, 360, 392]. Most of these backdoors are forgotten if the compromised clients do not regularly participate in training, but the backdoor becomes more durable if injected in the lowest utilized model parameters [441].

Supply chain model poisoning. [**NISTAML.05**] [**NISTAML.051**] [Back to Index] Model poisoning attacks are also possible in supply-chain scenarios in which models or components of the model provided by suppliers are poisoned with malicious code. Dropout Attack [425] is a recent supply-chain attack that shows how an adversary who manipulates the randomness used in neural network training (particularly in dropout regularization) might poison the model to decrease accuracy, precision, or recall on a set of targeted classes. See Supply Chain Attacks and Mitigations for additional discussion of supply-chain risks to GenAI models that are applicable to PredAI models too.

Mitigations. A variety of Byzantine-resilient aggregation rules have been designed and evaluated to defend federated learning from model poisoning attacks. Most of them attempt to identify and exclude the malicious updates when performing the aggregation at the server [8, 43, 51, 149, 242–244, 359, 423]. However, motivated adversaries can bypass these defenses by adding constraints to the attack generation optimization problem [23, 123, 335]. Gradient clipping and differential privacy have the potential to mitigate model poisoning attacks to some extent [23, 271, 360], but they usually decrease accuracy and do not provide complete mitigation.

For specific model poisoning vulnerabilities, such as backdoor attacks, there are some techniques for model inspection and sanitization (see Sec. 2.3.3). However, mitigating supply-chain attacks in which adversaries might control the source code of the training algorithm or the ML hyperparameters remains challenging. Program verification techniques used in other domains (e.g., cryptographic protocol verification [299]) might be adapted to this setting, but ML algorithms have intrinsic randomness and non-deterministic behavior, which enhances the difficulty of verification.

> Designing ML models that are robust in the face of supply-chain model poisoning vulnerabilities is a critical open problem.

2.3.5. Poisoning Attacks in the Real World

As poisoning attacks require adversarial control over the ML training process, they are difficult to mount in the real world. Still, there are several examples of documented cases of real poisoning attacks targeting early AI chatbots, email spam filters, and malware classification services.

The first example of a real-world poisoning attack is the Tay.AI chatbot, a chatbot released by Microsoft on Twitter in 2016 [272]. After online interaction with users for less than 24 hours, the chatbot was poisoned and immediately taken down. At about the same time, there were several large-scale efforts to compromise Google's Gmail spam filter, in which attackers sent millions of emails to attempt to poison the Gmail spam classifier algorithm, enabling them to send other malicious emails without being detected [272]. MITRE ATLAS reported a poisoning incident on the VirusTotal threat intelligence service, in which similar, but not identical samples of a ransomware family were submitted through a popular virus sharing platform to cause the mis-classification of that particular ransomware family [248].

These incidents highlight the risks associated with online learning, as the Tay.AI chatbot was updated in real-time based on user interactions, and the Gmail spam filter and the VirusTotal malware classification system were continuously updated based on newly received samples. In all these incidents, attackers crafted poisoned samples after an initial model release, counting on the fact that models are continuously updated.

2.4. Privacy Attacks and Mitigations

[**NISTAML.03**] [Back to Index]

The seminal work of Dinur and Nissim [110] introduced DATA RECONSTRUCTION attacks, which seek to reverse-engineer private information about an individual user record or other sensitive input data from access to a trained model. More recently, data reconstruction attacks have been designed for binary and multi-class neural network classifiers [50, 152]. With MEMBERSHIP-INFERENCE ATTACK, an adversary can determine whether a particular record was included in the dataset used for training an ML model. Membership inference attacks were first introduced by Homer et al. [162] for genomic data. Recent literature focuses on membership attacks against ML models in mostly black-box settings in which adversaries have query access to a trained ML model [54, 342, 422]. Property inference attacks [19, 74, 134, 233, 361, 437] aim to extract global information about a training dataset, such as the fraction of training examples with a certain sensitive attribute. A different privacy violation for MLaaS is MODEL EXTRACTION attacks, which are designed to extract information about an ML model, such as its architecture or model parameters[3] [58, 70, 177, 376].

This section discusses privacy attacks related to data reconstruction, the memorization of training data, membership inference, property inference, and model extraction, as well as mitigations for some of these attacks and open problems in designing general mitigation strategies.

2.4.1. Data Reconstruction

[**NISTAML.032**] [Back to Index]

Data reconstruction attacks have the ability to recover an individual's data from released aggregate information. Dinur and Nissim [110] were the first to introduce reconstruction attacks that recover user data from linear statistics. Their original attack required an exponential number of queries for reconstruction, but subsequent work has shown how to perform reconstruction with a polynomial number of queries [116]. A survey of privacy attacks, including reconstruction attacks, is given by Dwork et al. [114]. More recently, the U.S. Census Bureau performed a large-scale study on the risk of data reconstruction attacks on census data [135], which motivated the use of differential privacy in the decennial release of the U.S. Census in 2020.

In the context of ML classifiers, Fredrickson et al. [130] introduced model inversion attacks that reconstruct class representatives from the training data of an ML model. While

[3]A privacy violation in this context describes a loss of confidential information about an ML model. If ML model leakage leads to further privacy violations for individuals (e.g., identity theft, sensitive data extraction), it may also be viewed as a cybersecurity-related privacy event. For further discussion on the relationship between privacy and cybersecurity risks, see NIST Privacy Framework, Version 1.0.

model inversion generates semantically similar images as those in the training set, it cannot directly reconstruct the training data of the model. Recently, Balle et al. [26] trained a reconstructor network that can recover a data sample from a neural network model, assuming that a powerful adversary has information about all other training samples. Haim et al. [152] showed how the training data of a binary neural network classifier can be reconstructed from access to the model parameters by leveraging theoretical insights about implicit bias in neural networks. This work has recently been extended to reconstruct training samples of multi-class multi-layer perceptron classifiers [50]. Attribute inference is another relevant privacy attack in which the attacker extracts a sensitive attribute of the training set, assuming partial knowledge about other features in the training data [184].

The ability to reconstruct training samples is partially explained by the tendency of neural networks to memorize their training data. Zhang et al. [431] discussed how neural networks can memorize randomly selected datasets. Feldman [126] showed that the memorization of training labels is necessary to achieve an almost optimal generalization error in ML. Brown et al. [48] constructed two learning tasks based on next-symbol prediction and cluster labeling in which memorization is required for high-accuracy learning. Feldman and Zhang empirically evaluated the benefit of memorization for generalization using an influence estimation method [127]. Data reconstruction attacks and their connection to memorization for generative AI are discussed in Sec. 3.3.2.

2.4.2. Membership Inference

[**NISTAML.033**] [Back to Index]

Membership inference attacks may expose private information about an individual, like reconstruction or memorization attacks, and are of great concern when releasing aggregate information or ML models trained on user data. In certain situations, determining that an individual is part of the training set already has privacy implications, such as in a medical study of patients with a rare disease. Moreover, membership inference can be used as a building block for mounting data extraction attacks [59, 63].

In membership inference, the attacker's goal is to determine whether a particular record or data sample was part of the training dataset used for the statistical or ML algorithm. These attacks were introduced by Homer et al. [162] for statistical computations on genomic data under the name *tracing attacks*. Robust tracing attacks have been analyzed when an adversary gains access to noisy statistical information about the dataset [115]. In the last five years, the literature has used the terminology *membership inference* for attacks against ML models. Most of the attacks in the literature are performed against deep neural networks that are used for classification [54, 89, 208, 342, 421, 422]. Similar to other attacks in AML, membership inference can be performed in white-box settings [208, 264, 317] in which attackers have knowledge of the model's architecture and parameters, but most of the attacks have been developed for black-box settings in which the adversary generates queries to the trained ML model [54, 89, 342, 421, 422].

The attacker's success in membership inference has been formally defined using a cryptographically inspired privacy game in which the attacker interacts with a challenger and needs to determine whether a target sample was used in training the queried ML model [183, 321, 422]. In terms of techniques for mounting membership inference attacks, the loss-based attack by Yeom et al. [422] is one of the most efficient and widely used method. Using the knowledge that the ML model minimizes the loss on training samples, the attack determines that a target sample is part of training if its loss is lower than a fixed threshold (selected as the average loss of training examples). Sablayrolles et al. [317] refined the loss-based attack by scaling the loss using a per-example threshold. Another popular technique introduced by Shokri et al. [342] is *shadow models*, which trains a meta-classifier on examples in and out of the training set obtained by training thousands of shadow ML models on the same task as the original model. This technique is generally expensive, and while it might improve upon the simple loss-based attack, its computational cost is high and requires access to many samples from the distribution to train the shadow models. These two techniques are at opposite ends of the spectrum in terms of their complexity, but they perform similarly in terms of precision at low false positive rates [54].

An intermediary method that obtains good performance in terms of the AREA UNDER THE CURVE (AUC) metric is the LiRA attack by Carlini et al. [54], which trains a smaller number of shadow models to learn the distribution of model logits on examples in and out of the training set. Using the assumption that the model logit distributions are Gaussian, LiRA performs a hypothesis test for membership inference by estimating the mean and standard deviation of the Gaussian distributions. Ye et al. [421] designed a similar attack that performs a one-sided hypothesis test, which does not make any assumptions on the loss distribution but achieves slightly lower performance than LiRA. Recently, Lopez et al. [225] proposed a more efficient membership inference attack that requires training a single model to predict the quantiles of the confidence score distribution of the model under attack. Membership inference attacks have also been designed under the stricter label-only threat model in which the adversary only has access to the predicted labels of the queried samples [89].

There are several public privacy libraries that offer implementations of membership inference attacks: the TensorFlow Privacy library [350] and the ML Privacy Meter [259].

2.4.3. Property Inference

[**NISTAML.034**] [Back to Index]

In property inference attacks (also called distribution inference), the attacker tries to learn global information about the training data distribution by interacting with an ML model. For example, an attacker can determine the fraction of the training set with a certain sensitive attribute (e.g., demographic information) that might reveal potentially confidential information about the training set that is not intended to be released.

Property inference attacks were introduced by Ateniese et al. [19] and formalized as a distinguishing game between the attacker and the challenger training two models with different fractions of the sensitive data [361]. Property inference attacks were designed in white-box settings in which the attacker has access to the full ML model [19, 134, 361] and black-box settings in which the attacker issues queries to the model and learns either the predicted labels [233] or the class probabilities [74, 437]. These attacks have been demonstrated for HIDDEN MARKOV MODEL, SUPPORT VECTOR MACHINES [19], FEEDFORWARD NEURAL NETWORKS [134, 233, 437], CONVOLUTIONAL NEURAL NETWORKS [361], FEDERATED LEARNING [240], GENERATIVE ADVERSARIAL NETWORKS [443], and GRAPH NEURAL NETWORK [442]. Mahloujifar et al. [233] and Chaudhauri et al. [74] showed that poisoning the property of interest can help design a more effective distinguishing test for property inference. Moreover, Chaudhauri et al. [74] designed an efficient property size estimation attack that recovers the exact fraction of the population of interest.

The relationship between different training set inference attacks, such as membership inference, attribute inference, and property inference, has been explored by Salem et al.[321] under a unified definitional framework.

2.4.4. Model Extraction

[**NISTAML.031**] [Back to Index]

In MLaaS scenarios, cloud providers typically train large ML models using proprietary data and would like to keep the model architecture and parameters confidential. The goal of an attacker performing 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 by Tramer at al. [376] on several online ML services for different ML models, including logistic regression, decision trees, and neural networks. However, Jagielski et al. [177] have shown the exact extraction of ML models to be impossible. Instead, a functionally equivalent model can be reconstructed that is different than the original model but achieves similar performance at the prediction task. Jagielski et al. [177] have shown that even the weaker task of extracting functionally equivalent models is computationally prohibitive (*NP*-hard).

Several techniques for mounting model extraction attacks have been introduced in the literature. The first method is that of direct extraction based on the mathematical formulation of the operations performed in deep neural networks, which allows the adversary to compute model weights algebraically [58, 177, 376]. A second technique is to use learning methods for extraction. For example, active learning [70] can guide the queries to the ML model for more efficient extraction of model weights, and reinforcement learning can train an adaptive strategy that reduces the number of queries [280]. A third technique uses SIDE CHANNEL information for model extraction. Batina et al. [29] used electromagnetic side channels to recover simple neural network models, while Rakin et al. [303] showed how ROWHAMMER ATTACK can be used for model extraction of more complex convolutional neural network architectures.

Model extraction is often not an end goal but a step toward other attacks. As the model weights and architecture become known, attackers can launch more powerful attacks that are typical for the white-box or gray-box settings. Therefore, preventing model extraction can mitigate downstream attacks that depend on the attacker having knowledge of the model architecture and weights.

2.4.5. Mitigations

The discovery of reconstruction attacks against aggregate information motivated the rigorous definition of *differential privacy* (DP) [112, 113], an extremely strong definition of privacy that guarantees a bound on how much an attacker with access to the algorithm output can learn about each individual record in the dataset. The original *pure* definition of DP has a privacy parameter ε (i.e., privacy budget), which bounds the probability that the attacker with access to the algorithm's output can determine whether a particular record was included in the dataset. DP has been extended to the notions of approximate DP, which includes a second parameter δ that is interpreted as the probability of information accidentally being leaked in addition to ε and Rènyi DP [246].

DP has been widely adopted due to several useful properties: group privacy (i.e., the extension of the definition to two datasets that differ in k records), post-processing (i.e., privacy is preserved even after processing the output), and composition (i.e., privacy is composed if multiple computations are performed on the dataset). DP mechanisms for statistical computations include the Gaussian mechanism [113], the Laplace mechanism [113], and the Exponential mechanism [238]. The most widely used DP algorithm for training ML models is DP-SGD [1], and recent improvements include DP-FTRL [189] and DP matrix factorization [105].

By definition, DP provides mitigation against data reconstruction and membership inference attacks. In fact, the definition of DP immediately implies an upper bound on the success of an adversary in mounting a membership inference attack. Tight bounds on the success of membership inference have been derived by Thudi et al. [369]. However, DP does not provide guarantees against model extraction attacks, as this method is designed to protect the training data, not the model. Several papers have reported negative results after using differential privacy to protect against property inference attacks that aim to extract the properties of subpopulations in the training set [74, 233].

One of the main challenges of using DP in practice is setting up the privacy parameters to achieve a trade-off between the level of privacy and the achieved utility, which is typically measured in terms of accuracy for ML models. Analysis of privacy-preserving algorithms (e.g., DP-SGD) is often worst-case and not tight, and selecting privacy parameters based purely on theoretical analysis results in utility loss. Therefore, large privacy parameters are often used in practice (e.g., the 2020 U.S. Census release used $\varepsilon = 19.61$), and the exact privacy obtained in practice is difficult to estimate. Jagielski et al. [181] introduced *privacy auditing* with the goal of empirically measuring the actual privacy guarantees of an algorithm and determining privacy lower bounds by mounting privacy attacks. Many privacy auditing techniques are based on inserting *canaries* – synthetic and easy-to-identify out-of-distribution examples – into the training set, and then measuring the canary presence in model output. Auditing can also be performed with membership inference attacks [183, 427], but intentional insertion of strong canaries may result in better estimates of privacy leakage [181, 265]. Recent advances in privacy auditing include tighter bounds for the Gaussian mechanism [263] and rigorous statistical methods that allow for the use of multiple canaries to reduce the sample complexity of auditing [297]. Additionally, two efficient methods for privacy auditing with training a single model have been proposed: Steinke et al. [355] use, multiple random data canaries without incurring the cost of group privacy; and Andrew et al. [10] used multiple random client canaries and cosine similarity test statistics to audit user-level private federated learning.

> Differential privacy provides a rigorous notion of privacy and protects against membership inference and data reconstruction attacks. To achieve the best balance between privacy and utility, empirical privacy auditing is recommended to complement the theoretical analysis of private training algorithms.

There are other mitigation techniques against model extraction, such as limiting user queries to the model, detecting suspicious queries to the model, or creating more robust architectures to prevent side-channel attacks. However, these techniques can be circumvented by motivated and well-resourced attackers and should be used with caution. There are practice guides available for securing ML deployments [69, 274]. A completely different approach to potentially mitigating privacy leakage of a user's data is to perform MACHINE UNLEARNING, a technique that enables a user to request the removal of their data from a trained ML model. Existing techniques for machine unlearning are either exact (i.e., retraining the model from scratch or from a certain checkpoint) [45, 52] or approximate (i.e., updating the model parameters to remove the influence of the unlearned records) [139, 175, 268]. They offer different tradeoffs between computation and privacy guarantees, with exact unlearning methods offering stronger privacy, at additional computational cost.

3. Generative AI Taxonomy

GenAI is a branch of AI that develops models that can generate content (e.g., images, text, and other media) with similar properties to their training data. GenAI includes several different types of AI technologies with distinct origins, modeling approaches, and related properties, including: GENERATIVE ADVERSARIAL NETWORKS, GENERATIVE PRE-TRAINED TRANSFORMER (GPT), and DIFFUSION MODELS, among others. Recently, GenAI systems have emerged with multi-modal content generation or comprehension capabilities [119], sometimes through combining two or more model types.

3.1. Attack Classification

While many attack types in the PredAI taxonomy apply to GenAI (e.g., data poisoning, model poisoning, and model extraction), recent work has also introduced novel AML attacks specific to GenAI systems.

Figure 2 shows a taxonomy of attacks in AML for GenAI systems. Similar to the PredAI taxonomy in Fig. 1, this taxonomy is first categorized by the system properties that attackers seek to compromise in each case, including **availability** breakdowns, **integrity** violations, and **privacy** compromises, as well as the additional category of AML attack relevant to GenAI of **misuse** enablement, in which attackers seek to circumvent restrictions placed on the outputs of GenAI systems (see Sec. 2.1.2). The capabilities that an adversary must leverage to achieve their objectives are shown in the outer layer of the objective circles. Attack classes are shown as callouts connected to the capabilities required to mount each attack. Where there are specific types of a more general class of attack (for example, a jailbreak is a specific kind of direct prompting attack attack), the specific attack is linked to the more general attack class through an additional callout. Certain attack classes are listed multiple times because the same attack technique can be used to achieve different attacker goals.

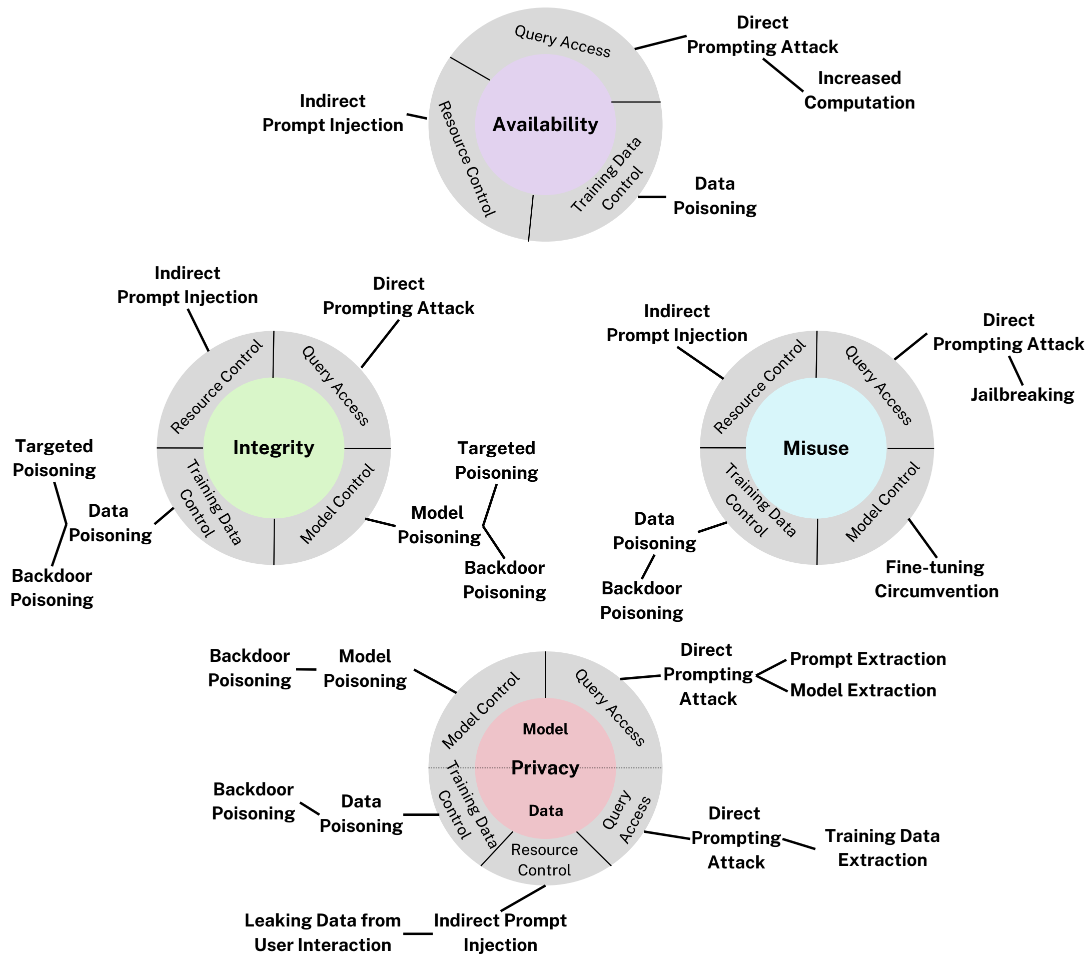

Figure 2. Taxonomy of attacks on GenAI systems

An attack can be further categorized by the learning stage to which it applies and by the attacker's knowledge and access. These are reviewed in the following sections. Where possible, the discussion broadly applies to GenAI models, though some attacks may be most relevant to particular kinds of GenAI models or model-based systems such as RETRIEVAL-AUGMENTED GENERATION (RAG) [RAG] systems, chatbots, or AGENT systems.

3.1.1. GenAI Stages of Learning

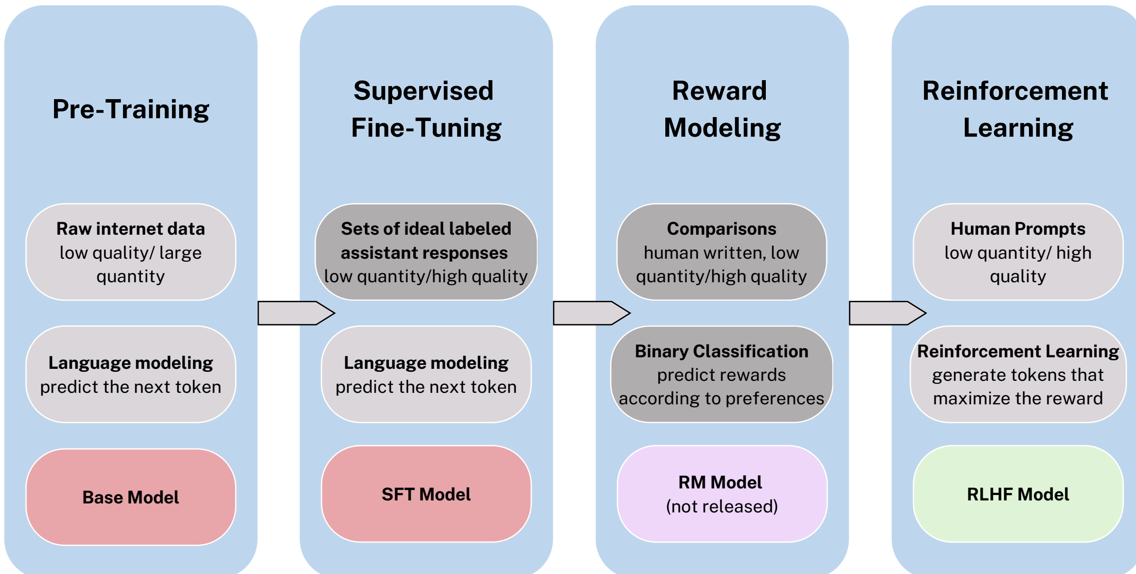

Figure 3. Example LLM Training Pipeline used for InstructGPT [281]

The GenAI development pipeline shapes the space of possible AML attacks against GenAI models and systems. In GenAI moreso than in PredAI, different activities such as data collection, model training, model deployment, and application development are often carried out by multiple different organizations or actors.

For example, a common paradigm in GenAI is the use of a smaller number of foundation models to support a diverse range of downstream applications. Foundation models are pre-trained on large-scale data using self-supervised learning in order to encode general patterns in text, images, or other data that may be relevant for many different applications [311]. Data at the scale used in foundation models is often collected from a variety of internet sources (which attackers can target, such as in DATA POISONING attacks).

This generalist learning paradigm equips foundation models with a variety of capabilities and tendencies — many of which are desirable, but some of which may be harmful or unwanted by the model developer. Techniques such as supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) can be used after initial pre-training to better align the base model with human preferences and to curb undesirable or harmful model outputs [281] (see Fig. 3). However, these interventions can later be targeted using AML techniques by attackers seeking to recover or re-enable potentially harmful capabilities.

Developers can make trained foundation models available to downstream users and developers in a variety of ways, including openly releasing the model's weights for re-use and

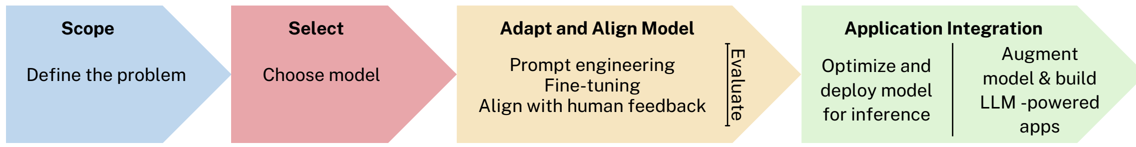

Figure 4. LLM enterprise adoption pipeline

modification, or hosting the model and offering access as a service through an API. These release decisions impact attacker capabilities that shape the space of possible AML attacks, such as whether attackers possess MODEL CONTROL.

Depending on how a foundation model has been made available, downstream developers can customize and build upon the model to create new applications, such as by further fine-tuning the model for a specific use case, or by integrating a foundation model with a software system, such as to build a retrieval-augmented generation (RAG) or agent (see Figure 4). Thus, a foundation model's vulnerabilities to AML attacks can potentially impact a wide range of downstream applications and end users. At the same time, the specific application context in which a foundation model is integrated can create additional vectors for and risks from AML attacks, such as the potential exposure of application-specific data.

AML attacks differ and depend on different phases of the GenAI development lifecycle. One major division is between attacks that target the training stage and those that target model inference during the deployment stage.

Training-time attacks. [**NISTAML.037**] [Back to Index] The TRAINING STAGE for GenAI often consists of foundation model PRE-TRAINING and model FINE-TUNING. This pattern exists for generative image models, text models, audio models, and multimodal models, among others. Since foundation models are most effective when trained on large datasets, it has become common to scrape data from a wide range of public sources, increasing the vulnerability of these models to DATA POISONING attacks. Additionally, GenAI systems trained or fine-tuned by third parties are often used in downstream applications, leading to the risk of MODEL POISONING attacks from maliciously constructed models.

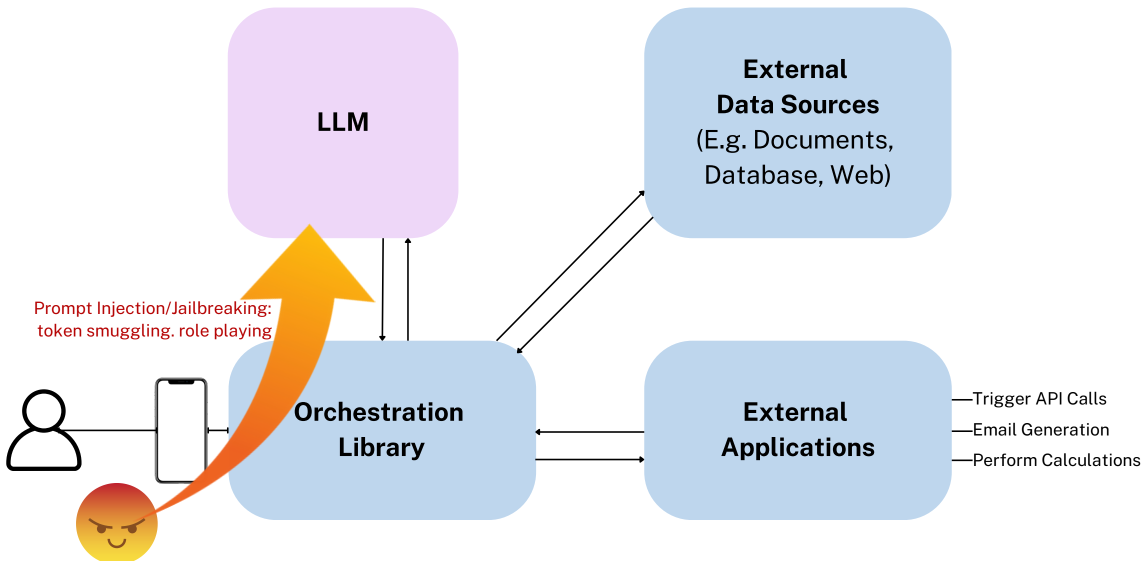

Figure 5. LLM enterprise adoption reference architecture

Inference-time attacks. The DEPLOYMENT STAGE for GenAI models and systems varies both based on how models are hosted or otherwise made available to users, and in how they are integrated into downstream applications. However, GenAI models and applications often share properties that leave them vulnerable to similar types of attacks. For example, underlying many of the security vulnerabilities in LLM applications is the fact that data and instructions are not provided in separate channels to the LLM, which allows attackers to use data channels to inject malicious instructions in inference-time attacks (a similar flaw to that which underlies decades-old SQL injection attacks). Many of the attacks in this stage are due to the following practices that are common in applications of text-based generative models:

1. **In-context instructions and system prompts:** [**NISTAML.035**] [Back to Index] The behavior of LLMs can be shaped through inference-time prompting, whereby the developer or user provides in-context instructions that are often prepended to the model's other input and context. These instructions comprise a natural language description of the model's application-specific use case (e.g., "You are a helpful financial assistant who responds gracefully and concisely....") and is known as a SYSTEM PROMPT. A PROMPT INJECTION overrides these instructions, exploiting the concatenation of untrusted user output to the system prompt to induce unintended behavior. For example, an attacker could inject a JAILBREAK that overrides the system prompt to cause the model to generate restricted or unsafe outputs. Since these prompts have been carefully crafted through prompt engineering and may be security-relevant, a PROMPT EXTRACTION attack may attempt to steal these system instructions. These attacks are also relevant to multimodal and text-to-image models.

2. **Runtime data ingestion from third-party sources:** In RETRIEVAL-AUGMENTED GENERA-

TION (RAG) applications, chatbots, and other applications in which GenAI models are used to interface with additional resources, context is often crafted at runtime in a query-dependent way and populated from external data sources (e.g., documents, web pages, etc.) that are to be used as part of the application. INDIRECT PROMPT INJECTION attacks depend on the attacker's ability to modify external sources of information that will be ingested into the model context, even if not provided directly by the primary system user.

3. **Output handling:** The output of an GenAI model may be used dynamically, such as to populate an element on a web page or to construct a command that is executed without any human supervision, which can lead to a range of availability, integrity or privacy violations in downstream applications if an attacker can induce behavior in this output that the developer has not accounted for.

4. **Agents:** An LLM-based AGENT relies on iteratively processing the *output* of an LLM (item 3 above) to perform a task and then provides the results as additional context back to the LLM input (item 2) [151, 155, 393]. For example, an agent system may select from among a configured set of external dependencies and invoke the code with templates filled out by the LLM using information in the context. Adversarial inputs into this context, such as from interactions with untrusted resources, could hijack the agent into performing adversary-specified actions instead, leading to potential security or safety violations.

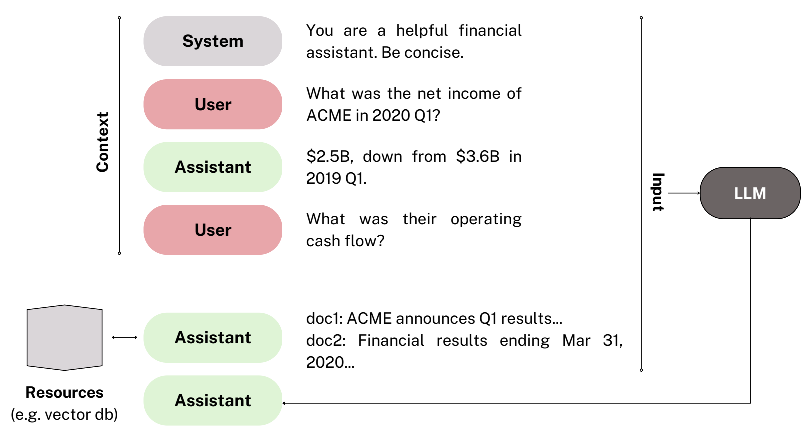

Figure 6. Retrieval-augmented generation

3.1.2. Attacker Goals and Objectives

As with PredAI, attacker objectives can be classified broadly along the dimensions of availability, integrity, and privacy, along with a new, GenAI-specific category of attacks designed to enable misuse.

- In an **AVAILABILITY BREAKDOWN attack** [**NISTAML.01**] [Back to Index], an attacker seeks to interfere with a GenAI model or system to disrupt the ability of other users or processes to obtain timely and consistent access to its outputs or functionality.

- In an **INTEGRITY VIOLATION attack** [**NISTAML.02**] [Back to Index], an attacker seeks to interfere with a GenAI system to force it to misperform against its intended objectives and produce output that aligns with the attacker's objective. As users and businesses rely on GenAI systems to perform tasks like research and productivity assistance, these violations can allow attackers to weaponize the trust that these users place in GenAI systems.

- In a **PRIVACY COMPROMISE attack** [**NISTAML.03**] [Back to Index], an attacker seeks to gain unauthorized access to restricted or proprietary information that is part of a GenAI system, including information about a model's training data, weights or architecture; or sensitive information that the model accesses such as the knowledge base of a RETRIEVAL-AUGMENTED GENERATION (RAG) application. GenAI systems may be exposed to sensitive data (intentionally or otherwise) during training or inference, and attacks may seek to extract such information (e.g., through INDIRECT PROMPT INJECTION attacks, where a third party exfiltrates in-context user information [307], or a MODEL EXTRACTION attack to exfiltrate model information [61]).

- **Misuse enablement** [**NISTAML.04**][Back to Index]. An additional attacker objective that is especially relevant in the GenAI context is the goal of **MISUSE ENABLEMENT**. In these attacks, an attacker seeks to deliberately circumvent technical restrictions imposed by the GenAI system's owner on its use, such as restrictions designed to prevent the system from producing outputs that could cause harm to others, cf. [325].

  Technical restrictions refer in this context to defenses applied to the GenAI system such as the use of system prompts or RLHF for safety alignment. While the specific technical restrictions in place will vary between models, the techniques for circumventing such defenses are often common between different kinds of models and different kinds of misuse, allowing them to be taxonomized as a part of AML without specificity as to the particular kinds of misuse that model developers seek to prevent.

3.1.3. Attacker Capabilities

AML attacks can be taxonomized with respect to the capabilities that an attacker has in controlling inputs to the GenAI model or system. These capabilities include:

- TRAINING DATA CONTROL: The attacker might take control of a subset of the training data by inserting or modifying training samples. This capability is used in DATA POISONING attacks.

- QUERY ACCESS: Many GenAI models and their applications are deployed as services that can be accessed over the internet by users. In these cases, attackers can submit adversarially crafted queries to the model to elicit specific desired behavior or extract information. This capability is used for PROMPT INJECTION, PROMPT EXTRACTION, and MODEL EXTRACTION attacks. Query access can vary based on the degree of generation control (e.g., modifying the temperature or adding a logit bias) and the richness of the returned generation (e.g., with or without log probabilities or multiple choices).

- RESOURCE CONTROL: The attacker might modify resources (e.g., documents, web pages) that will be ingested by the GenAI model at runtime. This capability is used for INDIRECT PROMPT INJECTION attacks.
- MODEL CONTROL: The attacker might have the ability to modify model parameters, such as through public fine-tuning APIs or openly accessible model weights. This capability is used in MODEL POISONING attacks, as well FINE-TUNING CIRCUMVENTION attacks which remove refusal behavior or other model-level safety interventions [132, 153, 300].

As in PredAI, attackers can also vary in their knowledge of the underlying ML model, from full knowledge of the ML system including model weights (**white-box attacks**), to minimal knowledge and systems with deliberately obscured or misleading information (**black-box attacks**), to somewhere in between (**gray-box attacks**). See Sec. 2.1.4, which discusses attacker knowledge in greater detail and applies to GenAI attacks.

3.2. Supply Chain Attacks and Mitigations

[**NISTAML.05**] [Back to Index]

Since AI is software, it inherits many of the vulnerabilities of the traditional software supply chain, such as reliance on third-party dependencies. AI development also introduces new types of dependencies, including data collection and scoring, the integration or adaptation of third-party-developed AI models, and the integration of third-party-developed plugins into AI systems. Mitigating the security challenges in AI supply chain management is complex and requires a multifaceted approach that combines existing practices for software supply chain risk management with the management of AI-specific supply chain risks, such as through the use of provenance information for the additional artifacts involved [159, 267]. Studies of real-world security vulnerabilities against ML suggest that security is best addressed comprehensively and by considering the full attack surface, including data and model supply chains, software, and network and storage systems [17, 370]. While all of these supply chain risks are critical in the broader context of securing AI systems, there are certain types of attacks that rely on exploiting the specific statistical and data-based properties of ML systems, thus falling within the domain of AML.

3.2.1. Data Poisoning Attacks

The performance of GenAI text-to-image and text-to-text models has been found to scale with dataset size (among other properties like model size and data quality); for example, Hoffmann et al. [161] suggest compute-optimally training a 520 billion parameter model may require 11 trillion tokens of training data. Thus, it has become common for GenAI foundation model developers to scrape data from a wide range of sources. In turn, the scale of this data and the diversity of its sources provides a large potential attack surface into which attackers may seek to insert adversarially constructed data points. For example, dataset publishers may provide a list of URLs to constitute a training dataset, and attackers may be able to purchase some of the domains that serve those URLs and replace the site content with their own malicious content [57].

Beyond the vast quantities of pre-training data, data poisoning attacks may also affect other stages of the LLM training pipeline, including instruction tuning [389] and reinforcement learning from human feedback [305], which may intentionally source data from a large number of human participants.

As with PredAI models (see Sec. 2.1), data poisoning attacks could lead to attackers controlling model behavior through the insertion of a backdoor (see BACKDOOR POISONING ATTACK) such as a word or phrase that, when submitted to a model, acts as a universal JAILBREAK [305]. Attackers could also use data poisoning attacks to modify model behavior on particular user queries (see TARGETED POISONING ATTACK), such as causing the model to incorrectly summarize or otherwise produce degenerate outputs in response to queries that contain a particular trigger word or phrase [389]. These attacks may be practical—requiring a relatively small portion of the total dataset [46]—and may lead to a range of bad outcomes, such as code suggestion models which intentionally suggest insecure code [3].

3.2.2. Model Poisoning Attacks

[**NISTAML.051**] [Back to Index]

In GenAI, it is common for developers to use foundation models developed by third parties. Attackers can take advantage of this fact by offering maliciously designed models, such as pre-trained models that enable a BACKDOOR POISONING ATTACK or TARGETED POISONING ATTACK. While this attack relies on the attacker having control over the initial poisoned model, researchers have identified attacks in which malicious backdoors in pre-trained models can persist even after downstream users fine-tune the model for their own use [201] or apply additional safety training measures [170].

3.2.3. Mitigations

GenAI poisoning mitigations largely overlap with PredAI poisoning mitigations (see Sec. 2.3). For preventing data poisoning with web-scale data dependencies, this includes verifying web downloads as a basic integrity check to ensure that domain hijacking has not injected new sources of data into the training dataset [57]. That is, the provider publishes cryptographic hashes, and the downloader verifies the training data. Data filtering can also attempt to remove poisoned samples, though detecting poisoned data within a large training corpus may be very difficult.

While traditional software supply chain risk management practices such as vulnerability scanning of model artifacts can help manage some kinds of AI supply chain risks, new approaches are required to detect vulnerabilities in models such as those introduced through model poisoning attacks. Current proposed approaches include using methods from the field of mechanistic interpretability to identify backdoor features [67] and detecting and counteracting triggers when they are seen at inference time. Beyond these mitigations, risks can be reduced by understanding models as untrusted system components and designing applications such that risks from attacker-controlled model outputs are reduced [266].

3.3. Direct Prompting Attacks and Mitigations

[**NISTAML.018**] [Back to Index] DIRECT PROMPTING ATTACK attacks arise when the attacker is the primary user of the system, interacting with the model through query access. A subset of these attacks, in which the main user provides in-context instructions that are appended to higher-trust instructions like those provided by the application designer (such as the model's SYSTEM PROMPT), are known as DIRECT PROMPT INJECTION attacks.

As in PredAI, attacks may be applicable to a single setting and model, or may instead be universal (affecting models on a range of separate queries, see Sec. 2.2.1) and/or transferable (affecting models beyond the model they are found on, see Sec. 2.2.3).

An attacker may have a variety of goals when performing these attacks [219, 220, 337], such as to:

- **Enable misuse.** Attackers may use direct prompting attacks to bypass model-level defenses that a model developer or deployer has created to restrict models from producing harmful or undesirable output [237]. A JAILBREAK is a direct prompting attack intended to circumvent restrictions placed on model outputs, such as circumventing refusal behavior to enable misuse.
- **Invade privacy.** Attackers may use direct prompting to extract the system prompt or reveal private information that was provided to the model in context but not intended for unfiltered access by the user.
- **Violate integrity.** When LLMs are used as agents, an attacker may use direct prompting attacks to manipulate tool usage and API calls, and potentially compromise the backend of the system (e.g. executing attacker's SQL queries).

3.3.1. Attack Techniques

A range of techniques exist for launching direct prompting attacks, many of which generalise across various attacker objectives. With a focus on direct prompting attacks to enable misuse, we note the following broad categories of direct prompting techniques (see [393]):

- **Optimization-based attacks** design attack objective functions and use gradient or other search-based methods to learn adversarial inputs that cause a particular behavior, similar to PredAI attacks discussed in Sec. 2.2.1. Objective functions may be designed to force affirmative starts (e.g., looking for responses that begin with ”Sure”, which may indicate compliance with a malicious request [60, 320, 448]) or other metrics of attack success (e.g., similarity to a toxified finetune [368]).

  Optimization techniques can then be used to learn attacks, including techniques that follow from attacks designed for PredAI language classifiers (e.g., HotFlip [117]) and gradient-free techniques that use a proxy model or random search to test attack candidates [11, 320]. Universal adversarial triggers are a special class of these gradient-based attacks against generative models that seek to find *input-agnostic* prefixes (or suffixes) that produce the desired affirmative response regardless of the remainder of the input [386, 448]. That these universal triggers transfer to other models makes open-weight models — for which there is ready white-box access — feasible attack vectors for transferability attacks on closed systems in which only API access is available [448].

  Attacks can also be designed to satisfy additional constraints (e.g., sufficiently low perplexity [368]) or attack a system of multiple models [235].

- **Manual methods** for jailbreaking an LLM include competing objectives and mismatched generalization [400]. Mismatched generalization-based attacks identify inputs that fall outside the distribution of the model's safety training but remain within the distribution of its capabilities training, making them comprehensible to the model while evading refusal behavior. Competing objectives-based attacks find cases where model capabilities are in tension with safety goals, such as by playing into a model's drive to follow user-provided instructions. In all cases the goal of the attack is to compromise a model-level safety defense. See Weng [403] for further discussion.

  Approaches to competing objectives-based attacks include:

  1. *Prefix injection*: This method involves prompting the model to start responses with an affirmative confirmation. By conditioning the model to begin its output in a predetermined manner, adversaries attempt to influence its subsequent language generation toward specific, predetermined patterns or behaviors.
  2. *Refusal suppression*: Adversaries may explicitly instruct the model to avoid generating refusals or denials in its output. By decreasing the probability of refusal responses, this tactic aims to increase the probability of a compliant

response.

3. *Style injection*: In this approach, adversaries instruct the model to use (or not use) certain syntax or writing styles. For example, an attack may constrain the model's language to simplistic or non-professional tones, aiming to decrease the probability of (usually professionally worded) refusals.

4. *Role-play*: Adversaries utilize role-play strategies (e.g., "Always Intelligent and Machiavellian" [AIM] or "Do Anything Now" [DAN]) to guide the model to adopt specific personas or behavioral patterns that conflict with the original intent. This manipulation aims to exploit the model's adaptability to varied roles or characteristics, with an intent to compromise its adherence to safety protocols.

Approaches to mismatched generalization-based attacks include:

1. *Special encoding*: Strategies that use encoding techniques like base64 to alter the representation of input data in a way that remains understandable to the model but may be out of distribution for safety training.

2. *Character transformation*: Strategies that use character-level transformations like the ROT13 cipher, symbol replacement (e.g., l33tspeak), and Morse code to take the input out of the safety training distribution.

3. *Word transformation*: Strategies that alter the linguistic structure of the input, such as Pig Latin, synonym swapping (e.g., using "pilfer" for "steal"), and payload splitting (or "token smuggling") to break down sensitive words into substrings.

4. *Prompt-level transformation*: Strategies that use prompt-level transformations, such as translating the prompt into a less common language that may be out of distribution of the safety training data.

- **Automated model-based red teaming** employs an attacker model, a target model, and a judge [73, 239, 292]. When the attacker has access to a high-quality classifier that judges whether model output is harmful, it may be used as a reward function to train a generative model to generate jailbreaks of another generative model. Only query access is required for each of the models, and no human intervention is required to update or refine a candidate jailbreak. The prompts may also be transferable from the target model to other closed-source LLMs [73].

  The Crescendo attack [316] introduced the idea of interacting with the model iteratively in a multi-turn adaptive attack that includes seemingly benign prompts, eventually leading to a successful jailbreak against safety alignment. The initial manual attack is fully automated by leveraging another LLM for prompt generation and incorporating multiple input sources.

Evaluations of leading models suggest LLMs remain vulnerable to these many of these attacks [11, 338, 381].

3.3.2. Information Extraction

[**NISTAML.038**] [Back to Index] Both during training and at run-time, GenAI models are exposed to a range of information which may be of interest to attackers, like personally identifying information (PII) in the training data, sensitive information in RETRIEVAL-AUGMENTED GENERATION (RAG) databases provided in-context, or even the SYSTEM PROMPT constructed by the application designer. Additionally, features of the model itself—such as the model weights or architecture—may be targets of attack. Though many of the techniques in Sec. 3.3.1 apply to extracting such data, we note several specific goals and techniques specific to data extraction.

Leaking sensitive training data. Carlini et al. [59] were the first to practically demonstrate TRAINING DATA EXTRACTION attacks in generative language models. By inserting canaries–synthetic, easy-to-recognize out-of-distribution examples–in the training data, they developed a methodology for extracting the canaries and introduced a metric called *exposure* to measure memorization. Subsequent work demonstrated the risk of data extraction in LLMs based on transformers (e.g., GPT-2 [63]) by prompting the model with different prefixes and mounting a membership inference attack to determine which generated content was part of the training set. Since these decoder stack transformers are autoregressive models, a verbatim textual prefix about personal information can sometimes result in the model completing the text input with sensitive information that includes email addresses, phone numbers, and locations [229]. This behavior of verbatim memorization of sensitive information in GenAI language models has also been observed in more recent transformer models with the additional characterization of extraction methods [165]. Unlike PredAI models in which tools like Text Revealer are created to reconstruct text from transformer-based text classifiers [434], GenAI models can sometimes simply be asked to repeat private information that exists in the context as part of the conversation. Results show that information like email addresses can be revealed at rates exceeding 8% for certain models. However, their responses may wrongly assign the owner of the information and be otherwise unreliable. In general, extraction attacks are more successful when the model is seeded with more specific and complete information — the more the attacker knows, the more they can extract. Researchers have leveraged this fact to incrementally extract fragments of copyrighted New York Times articles from LLMs by seeding it with a single sentence, and allowing the LLM to recurrently extract additional text [356]. Intuitively, larger models with a higher capacity are more susceptible to exact reconstruction [56]. Fine-tuning interfaces also amplify the risk of data extraction attacks, as demonstrated by an attack that extracts PII from pre-training data using fine-tuning API for open-weight models [83], though this is not a direct prompting attack.

Prompt and context stealing. Prompts are vital to align LLMs to a specific use case and are

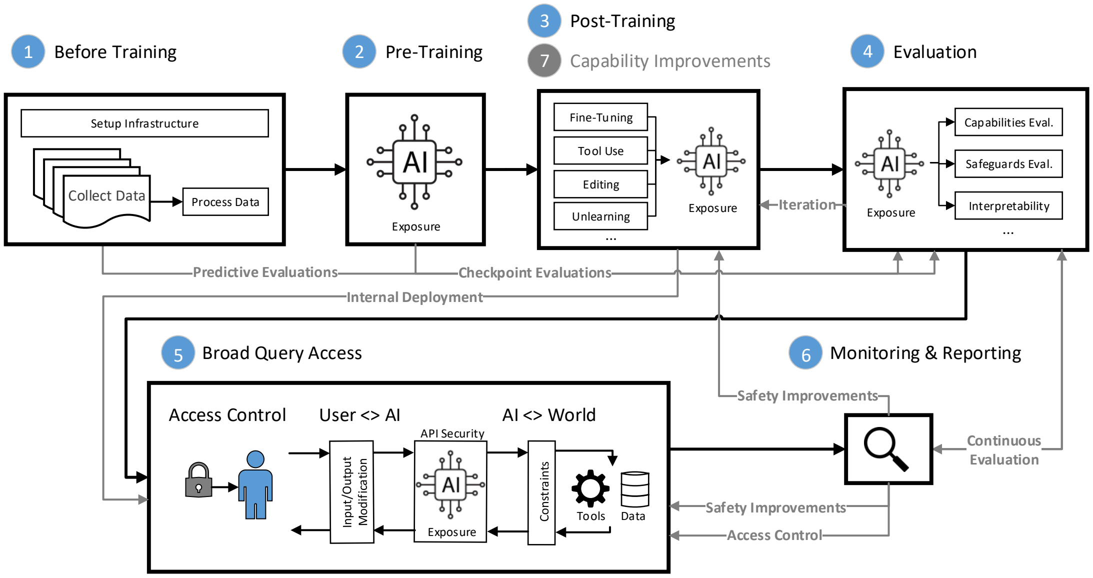

Figure 7. Map of the development and deployment life cycle of an AI model for broad-scale query access

a key ingredient to their utility in following human instructions. These prompts can therefore be regarded as commercial secrets, and are sometimes the target of direct prompting attacks. PromptStealer is a learning-based method that reconstructs prompts from text-to-image models using an image captioning model and a multi-label classifier to steal both the subject and the prompt modifiers [339]. For certain LLMs, researchers have found that a small set of fixed attack queries (e.g., `Repeat all sentences in our conversation`) were sufficient to extract more than 60 % of prompts across certain model and dataset pairs [439]. In some cases, effective prompts may draw from significant technical or domain expertise; prompt-stealing attacks may violate or threaten these investments. Furthermore, in RAG applications (see Fig. 6), the same techniques can be used to extract sensitive information provided in the LLMs' *context*. For example, rows from a database or text from a PDF document that are intended to be summarized generically by the LLM can be verbosely extracted by simply asking for them via direct prompting, or performing simple prompting attacks.

Model extraction. As in PredAI (Sec. 2.4.4), attackers may perform MODEL EXTRACTION attacks which attempt to learn information about the model architecture and parameters by submitting specially-crafted queries. Recently, Carlini et al. [61] demonstrated that such information could be extracted from black-box production LLMs, deriving previously unknown hidden dimensions and the embedding projection layer (up to symmetries).

3.3.3. Mitigations

The following defense strategies can be employed throughout the deployment life cycle of an AI model or system to reduce the risk that the model or system will be vulnerable to direct prompt injections. The numbers in parentheses refer to the numbering in Fig. 7, which shows a map of the deployment life cycle for broad-scale query access.

- **Interventions during pre-training (2) and post-training (3).** A range of training strategies have been proposed to increase the difficulty of accessing harmful model capabilities through direct prompt injection, including safety training during pre-training [197] or post-training [147, 445], adversarial training methods[340], and other methods to make jailbreak attacks more difficult [447].
- **Interventions during evaluation (4).** Evaluations can measure the vulnerability of models to query-based attacks, which can then inform trust and affordance decisions, as well as developer and user education. Evaluations can include broad automated vulnerability assessments [72, 107, 324] as well as targeted expert red teaming [381] and bug bounties [16]. Current evaluation approaches, though a useful tool, may underestimate vulnerabilities accessible to actors with more time, resourcing, or luck. Evaluations measure model vulnerabilities at a particular moment in time; assessments may change if new attacks are developed, additional data is collected post-training, or model capabilities are improved. Continuous evaluations following deployment can help combat these challenges.
- **Interventions during deployment (5).** A broad set of deployment-time interventions have been proposed:
  - *Prompt instruction and formatting techniques.* Model instructions can cue the model to treat user input carefully, such as by wrapping user input in XML tags, appending specific instructions to the prompt, or otherwise attempting to clearly separate system instructions from user prompts [14, 206, 219].
  - *Detecting and terminating harmful interactions.* Rather than preventing harmful model generations, AI systems may be able to detect these generations and terminate interactions. Several open [5, 6, 154] and closed [18, 204, 313] solutions have explored LLM-based detection systems with distinctly prompted and/or fine-tuned models that classify user input and/or model output as harmful or undesirable. These may provide supplementary assurance through a defense-in-depth philosophy. However, these detection systems are also vulnerable to attacks [235] and may have correlated failures to the main models that they are monitoring. Some lines of research have investigated constraining the space of generations to enable deterministic guardrails [306].Early work suggests that interpretability-based techniques can also be used to detect anomalous input [31], as well as keyword- or perplexity-based defenses [9, 164].

- *Prompt stealing detection.* A common approach to mitigating prompt stealing is to compare the model utterance to the prompt, which is known by the system provider. Defenses differ in how this comparison is made, which might include looking for a specific token, word, or phrase, as popularized by [59], or comparing the n-grams of the output to the input [439]. Similarly, defenses for prompt stealing have yet to be proven rigorous.
- *Input modification.* User input can additionally be modified prior to being passed to the model, such as paraphrasing or retokenizing [182]. However, such methods may be expensive and/or have trade-offs with model performance.
- *Aggregating output from multiple prompts.* Motivated by randomized smoothing [95] used to improve robustness against evasion attacks for ML classifiers, SmoothLLM [312] proposes aggregating the LLM output from multiple randomly perturbed prompts. This defense incurs a cost of generating multiple LLM queries for each prompt and might reduce the quality of the generated output.
- *Monitoring and response.* Following deployment, monitoring and logging of user activity may allow model deployers to identify and respond to instances of attempted and successful direct prompt injection attacks [266]. This response could include banning or otherwise acting against users if their intentions appear malicious, or remediating the prompt injection vulnerability in the event of a successful attack. Standard user- or organization-level vetting or identity verification procedures, as well as clear incentive mechanisms (such as a policy of restricting model access in response to violations) may enhance the efficacy of this mitigation.
- *Usage restrictions.* Other interventions have focused on choices about how models are offered to users: for example, the efficacy of some attacks can be reduced by limiting the inference parameters that are accessible to users (e.g., temperature or logit bias), as well as the richness of the model generations returned (e.g., logit probabilities) [250]. Additionally, limiting the release of public information [252, 266] and artifacts [249] and restricting the total number of model queries available to users [251] may make attacks more challenging. These techniques may have additional drawbacks in limiting positive use cases.

Indirect mitigations. Despite the growing number of proposed defenses at both the model and the system levels, recent findings suggest that current generation models remain highly vulnerable to direct prompt injection attacks [11, 381]. Thus, other potential mitigations for prompt injection rely not on directly increasing an AI system's robustness against such attacks, but instead on designing systems under the assumption that the AI model can and will produce malicious output if it is exposed to malicious actors. For example, deployers can design AI systems under the assumption that models with access to sensitive data or the ability to take undesirable actions may leak that data or take those actions [266]. Additionally, developers or deployers might use other technical mitigations to reduce the misuse potential of outputs obtained through direct prompt injection, such as:

- *Training data sanitization.* Model training data can be sanitized to remove sensitive or toxic content and data that are largely or exclusively relevant for developing undesirable capabilities. Such sanitization may prevent harmful capabilities from being learned and reduce the potential harms from direct prompt injection, though they may harm generalization and harmful content detection abilities [224].
- *Unlearning.* There have also been attempts to "unlearn" harmful knowledge or capabilities post-training [212], with the goal of reducing harms from maliciously directed models [158]. However, these methods remain vulnerable to adversarial attacks, including attacks on jailbreak-specific training approaches [367] and inversion attacks on unlearning methods [328], which extract supposedly unlearned data.
- *Watermarking.* Developers or deployers may watermark content generated by an AI model to help trace its provenance, distinguish it from human-generated content, and reduce risks from malicious use cases (e.g., by flagging content as model-generated when it appears online). While the literature has proposed various techniques with different strengths and weaknesses [194], there is no watermarking technique that is universally effective and robust under all circumstances. Many powerful attacks have been developed against watermarking with high success rates [188, 319]. Moreover, theoretical impossibility results regarding the robustness of watermarking have also been established [432].

Finally, beyond interventions at the developer or deployer levels, society and infrastructure can become more resilient to maliciously directed model capabilities over time [7, 33]. For example, defenders could adopt AI-based vulnerability discovery tools [99] to make their systems more resilient to malicious actors misusing GenAI models to find vulnerabilities for exploitation.

3.4. Indirect Prompt Injection Attacks and Mitigations

[**NISTAML.015] [Back to Index]** Many use cases for GenAI models involve models interacting with additional resources, from an internet-connected AGENT to a RETRIEVAL-AUGMENTED GENERATION (RAG) system depicted in Fig. 6. Because GenAI models combine the *data* and *instruction* channels, attackers can leverage the data channel to affect system operations by manipulating resources with which the system interacts. Thus, INDIRECT PROMPT INJECTION attacks are enabled by RESOURCE CONTROL that allows an attacker to indirectly (or remotely) inject system prompts without directly interacting with the application [146, 408]. Indirect prompt injection attacks can result in violations across at least three categories of attacker goals: 1) availability violation, 2) integrity violation, and 3) privacy compromise. However, unlike in direct prompt injection attacks, indirect prompt injection attacks are mounted not by the primary user of a model but instead by a third party. In fact, in many cases, it is the primary user of the model who is harmed by the compromise of the integrity, availability, or privacy of the GenAI system through an indirect prompt injection attack.

3.4.1. Availability Attacks

[**NISTAML.016] [Back to Index]** Attackers can manipulate resources to inject prompts into GenAI models that are designed to disrupt the availability of the model for legitimate users. Availability attacks can indiscriminately render a model unusable (e.g., failure to generate helpful outputs) or specifically block certain capabilities (e.g., specific APIs) [146].

Attacker techniques. Researchers have demonstrated several proof-of-concept methods by which attackers can disrupt the availability of a GenAI system:

- **Time-consuming background tasks.** [**NISTAML.017**] [Back to Index] An indirectly injected prompt can instruct the model to perform a time-consuming task prior to answering the request. The prompt itself can be brief, such as by requesting looping behavior in the evaluating model [146].
- **Inhibiting capabilities.** An indirectly injected prompt can instruct the model that it is not permitted to use certain APIs (e.g., the search API for an internet-connected chatbot). This selectively disarms key components of the service [146].
- **Disruptive output formatting.** An attacker can use indirect prompt injection to instruct the model to modify its output in a way that disrupts the availability of the system. For example, an attacker could instruct the model to replace the characters in retrieved text with homoglyph equivalents, disrupting subsequent API calls [146]; or could request that the model begins each sentence with an `<|endoftext|>` token, forcing the model to return an empty output [146].

3.4.2. Integrity Attacks

[**NISTAML.027] [Back to Index]**

Through indirect prompt injection, attackers can use malicious resources to prompt GenAI systems to become untrustworthy and generate content that deviates from benign behavior to align with adversarial objectives. These attacks often involve disrupting the model's behavior in subtle ways that may not be obvious to the end user.

For example, researchers have demonstrated attacks through indirect prompt injection that can cause a GenAI system to produce arbitrarily incorrect summaries of sources, to respond with attacker-specified information, or to suppress or hide certain information sources [146]. Attackers could use these capabilities to weaponize GenAI systems such as internet-connected chatbots against their users for a range of malign purposes, including spreading targeted misleading information, recommending fraudulent products or services, or redirecting consumers to malicious websites that spoof legitimate log-in pages or contain downloadable malware. Attackers may also use indirect prompt injection attacks to hijack a GenAI AGENT, causing it to perform a malicious, attacker-specified task instead of (or in addition to) its intended, user-provided task [353].

Attacker techniques. Researchers have demonstrated integrity attacks through malicious resources that manipulate the primary task of the LLM:

- **Jailbreaking.** Attackers can leverage techniques for indirect prompt injection that are similar to those used in direct prompt injection attacks, such as using a JAILBREAK that allows the attacker to substitute their own malicious instructions in place of the model's SYSTEM PROMPT. As in direct prompting attacks, these attacks may be crafted through optimization-based or manual methods, and may rely on techniques such as mismatched generalization.
- **Execution triggers.** Researchers have automated manual indirect prompt injection attacks using execution triggers generated via optimization with a technique called Neural Exec [287]. These execution triggers can also persist through RAG processing pipelines that include multiple phases, such as chunking and contextual filtering.
- **Knowledge base poisoning.** The knowledge database of a RAG system can be poisoned to achieved targeted LLM output to specific user queries, as in PoisonedRAG [449]. Recently, a general optimization framework called Phantom [75] has shown how a single poisoned document can be crafted and inserted into the knowledge database of a RAG system to induce a number of adversarial objectives in the LLM generator.
- **Injection hiding.** Attackers may use techniques to hide or obfuscate their injections, such as by hiding injections in non-visible portions of a resource; using multi-stage injections, in which the initial injection directs the model to visit another resource which contains additional injections; or encoding injection commands such as in Base64 and then instructing the model to decode the sequence[146].
- **Self-propagating injections.** Attackers may be able to use indirect prompt injection attacks to turn GenAI systems into vectors for spreading attacks. For example, an attacker could send a malicious email that, when read by a model integrated as part of an email client, instructs the model to spread the infection by sending similar malicious emails to everyone in the user's contact list. In this way, certain malicious prompts could serve as *worms* [146].

3.4.3. Privacy Compromise

Attackers can use indirect prompt injection attacks to compromise the privacy of a GenAI system or its primary users. For example, attackers could use indirect prompt injection attacks to compel a model to leak information from restricted resources, such as a user's private data that is processed by the GenAI system. Alternately, in a blend of integrity and privacy attacks, an attacker could gather information about the primary user or users of the system by instructing a model to obtain and then leak that information.

Attacker techniques. Researchers have theorized and demonstrated a variety of indirect prompt injection attacks to compromise information from internet-connected chatbots, RAG systems, and other GenAI systems. Some of these techniques include:

- **Compromising connected resources.** [**NISTAML.039] [Back to Index]** Attackers can use prompt injection attacks to cause a GenAI system to leak private information from the restricted resources it can access. For example, a model integrated as part of an email client could be prompted to forward certain emails to an attacker-controlled inbox [146]. Researchers have identified injection attacks that can force a model to exfiltrate user-uploaded data by querying an attacker-controlled URL with the sensitive data [298].
- **Leaking information from user interactions.** [**NISTAML.036] [Back to Index]** Researchers have demonstrated a proof-of-concept indirect prompt injection attack in which they inject instructions for a model to persuade the end user to reveal a piece of information (in this case, their name) that the model then leaks to the attacker, such as by directly querying an attacker-controlled URL with the information or suggesting such a URL to the user to visit [146]. Attackers may also be able to exploit features like markdown image rendering to exfiltrate data [323].

3.4.4. Mitigations

Various techniques (see Sec. 3.3.3) can be used throughout the development and deployment life cycle (Fig. 7) to mitigate attacks, including:

- Several training techniques have been developed to mitigate against indirect prompt injection, including fine-tuning task-specific models [296] and training models to follow hierarchical trust relationships in prompts [387].
- Detection schemes have been proposed to detect indirect prompt injection, and many LLM-based defenses have been designed to mitigate both direct and indirect prompt injection [6, 18, 154, 204, 313].
- A range of input processing methods have been proposed to combat indirect prompt injection, including filtering out instructions from third-party data sources [146], designing prompts to help aid LLMs in separating trusted and untrusted data (i.e., spotlighting [160, 206]), or instructing models to disregard instructions in untrusted data [206].

Many of the defenses described in the context of direct prompt injection can also be adapted to mitigate indirect prompt injection. Because current mitigations do not offer full protection against all attacker techniques, application designers may design systems with the assumption that prompt injection attacks are possible if a model is exposed to untrusted input sources, such as by using multiple LLMs with different permissions [145, 405] or by allowing models to interact with potentially untrustworthy data sources only through well-defined interfaces [410]. Additionally, public education efforts can inform model users and application designers of the risks of indirect prompt injection [266].

3.5. Security of Agents

An increasingly common use of GenAI models is constructing an (often LLM-based) AGENT, a software system that iteratively prompts a model, process its outputs – such as to select and call a function with specified inputs – and provides the results back to the model as a part of its next prompt [151, 155, 393]. Agents may be equipped to use tools such as web-browsing or code interpreters, and may have additional features such as memory and/or planning capabilities.

Because agents rely on GenAI systems to plan and execute their actions, they can be vulnerable to the many of the above categories of attacks against GenAI systems, including direct and indirect prompt injection. However, because agents can take actions using tools, these attacks can create additional risks in this context, such as enabling actors to hijack agents to execute arbitrary code or exfiltrate data from the environment in which they are operating. Security research focused specifically on agents is still in its early stages, but researchers have begun to evaluate the vulnerability of agents to particular AML attacks [12, 430] and to propose interventions to manage the security risks posed by agents [24].

3.6. Benchmarks for AML Vulnerabilities

There are several publicly available benchmarks for evaluating models' vulnerability to AML attacks. Datasets like JailbreakBench [72], AdvBench [448], HarmBench [237], StrongREJECT [351], AgentHarm [12], and Do-Not-Answer [399] provide benchmarks for evaluating models' susceptibility to jailbreaks. TrustLLM [169] is a benchmark intended to evaluate six dimensions of trust in LLMs: truthfulness, safety, fairness, robustness, privacy, and machine ethics. AgentDojo [101] is an evaluation framework for measuring the vulnerability of AI agents to prompt injection attacks in which the data returned by external tools hijacks the agent to execute malicious tasks. Additionally, open-source tools like Garak [106] and PyRIT [364] are intended to help developers identify vulnerabilities to AML attacks in models. Finally, several unlearning benchmarks have recently been proposed [212, 234].

4. Key Challenges and Discussion

4.1. Key Challenges in AML

There are several fundamental challenges that make fully addressing the problems discussed in this report more difficult. We discuss several here: The trade off between increasing accuracy (or average-case performance) and other attributes including robustness (or worst-case performance); the theoretical limitations and results that imply that fully robust systems may be mathematically impossible without additional assumptions; and the challenge of evaluating progress in AML mitigations rigorously and robustly.

4.1.1. Trade-Offs Between the Attributes of Trustworthy AI

The trustworthiness of an AI system depends on all of the attributes that characterize it [274]. There are trade-offs between explainability and adversarial robustness [176, 245] and between privacy and fairness [178]. For example, AI systems that are optimized for accuracy alone tend to underperform in terms of adversarial robustness and fairness [71, 111, 302, 379, 433]. Conversely, an AI system that is optimized for adversarial robustness may exhibit lower accuracy and deteriorated fairness outcomes [32, 391, 433]. Unfortunately, it may not be possible to simultaneously maximize the performance of an AI system with respect to these attributes.

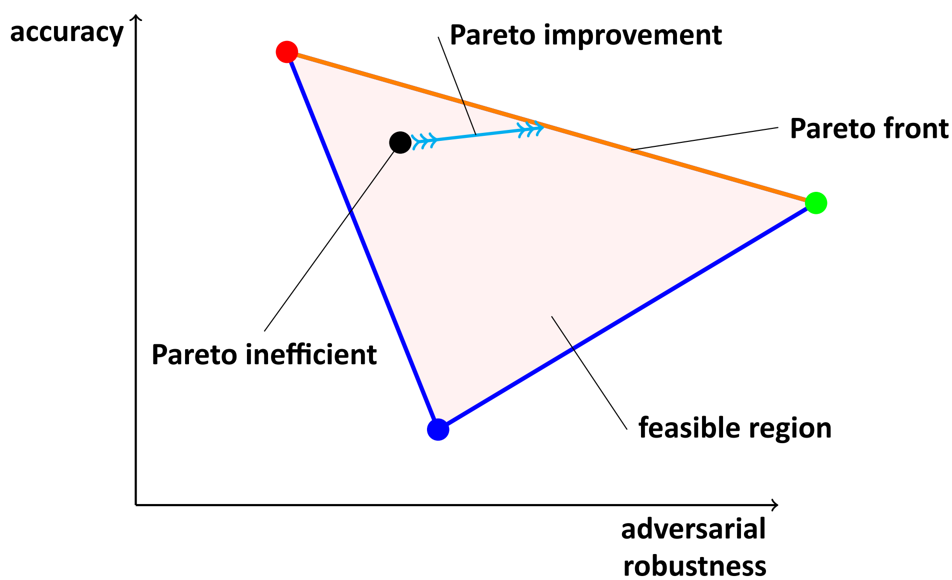

Figure 8. Pareto optimality

The full characterization of the trade-offs between the different attributes of trustworthy AI is an open research problem that is gaining importance with the adoption of AI technology in many areas of modern life. One promising practical approach is based on the concept of multi-objective optimization and Pareto optimality [285, 286]. In most cases, there is no mathematically best trade-off. However, Fig. 8 illustrates a hypothetical example of a trade-off between accuracy and adversarial robustness. Any point in the feasible region that is not on the Pareto front is a bad point (i.e., Pareto inefficient). There is a better solution (i.e., Pareto improvement) that can significantly help with one objective without harming the other, which is a goal of Pareto optimization. Moreover, if there is a single optimum in some use case, Pareto optimization naturally attains it. Organizations may need to accept trade-offs between these properties and decide which of them to prioritize depending on the AI system, the use case, and other relevant implications of the AI technology [274, 326, 382].

4.1.2. Theoretical Limitations on Adversarial Robustness

Given the multitude of powerful attacks, appropriate mitigations must be designed before AI systems are deployed in critical domains. This challenge is exacerbated by the lack of theoretically secure ML algorithms for many tasks in the field (see Sec. 1). This implies that designing mitigations is an inherently ad hoc and fallible process, though there are practice guides for securing ML deployments [69, 274] and existing guidelines for mitigating AML attacks [120].

An ongoing challenge in AML is the ability to detect when a model is under attack. Knowing this would provide an opportunity to counter the attack before any information is lost or an adverse behaviour is triggered in the model. However, Tramèr [373] has shown that designing techniques to detect adversarial examples is equivalent to robust classification, which is inherently difficult to solve. Adversarial examples may come from the same data distribution on which the model was trained and to which it expects the inputs to belong or may be OUT-OF-DISTRIBUTION (OOD) inputs. Thus, the ability to detect OOD inputs is also an important challenge in AML. Fang et al. [124] established useful theoretical bounds on detectability, particularly an impossibility result when there is an overlap between the in-distribution and OOD data.

Formal methods verification has a long history in other fields that require high assurance, such as avionics and cryptography. Although the results of applying this methodology offer security and safety assurances, they come at a very high cost, which has prevented formal methods from being widely adopted. Currently, formal methods in these fields are primarily used in applications that are mandated by regulations. Applying formal methods to neural networks has the potential to provide much-needed security guarantees, especially in high-risk applications. However, the viability of this technology will be determined by a combination of technical and business criteria — namely, the ability to handle today's complex ML models of interest at acceptable costs. More research is needed to extend this technology to the algebraic operations used in ML algorithms, scale it up to the large models used today, and accommodate rapid changes in the code of AI systems while limiting the costs of applying formal verification.

4.1.3. Evaluation

Another general problem of AML mitigations for both evasion and poisoning attacks is the lack of reliable benchmarks, which causes results from AML papers to be routinely incomparable, as they do not rely on the same assumptions and methods. While there have been some promising developments in this direction [97, 327], more research and encouragement are needed to foster the creation of standardized benchmarks to gain reliable insights into the actual performance of proposed mitigations.

More broadly, the effectiveness of a mitigation is determined not just by how well it will defend against existing attack, but also how well it defends against unforeseen attacks. This means that new mitigations should be tested adversarially, with the researchers proposing the mitigation also trying to break it. This is often difficult and time-consuming, leading to less rigorous and reliable evaluations of novel mitigations; often they appear very powerful, but are quickly shown lack robustness to unforeseen types of attacks.

Finally, this difficulty combines with the difficulty of trading off between different attributes discussed above. Instead of evaluating each attribute in isolation, they should be evaluated simultaneously for any new mitigations, and mitigations should be compared on a Pareto plot (as in Fig. 8) capturing the various tradeoffs that have to be made. This additionally increases the cost to evaluating new mitigations, and can make comparing mitigations difficult - if the green dot represents a new method, it is not possible to say it is an improvement on the red dot, as it is better on one axis but worse on the other.

4.2. Discussion

4.2.1. The Scale Challenge

Data is fundamentally important for training models. Recent trends in GenAI have been towards significant investment in larger models and larger datasets for training them. Few developers of foundation models publish key details about the data sources used in their training [44]. Those who do [247, 371] show the scale of the footprint and the massive amount of data consumed during training. The most recent multi-modal GenAI systems further exacerbate the demand by requiring large amounts of data for each modality.

In most cases, no single entity controls all of the data used to train a particular foundation model. Data repositories are not monolithic data containers but a list of labels and data links to other servers that actually contain the corresponding data samples. This paradigm challenges the classic definition of the corporate cybersecurity perimeter and creates new risks that are difficult to mitigate [57]. Recently published open-source data poisoning tools [241] increase the risk of large-scale attacks on image training data. Although created to enable artists to protect the copyright of their work, these tools may become harmful in the hands of people with malicious intent.

There are several ways this new class of attacks could be mitigated, although it is unclear how effective mitigations will prove to be as the sophistication of attacks increase. Data and model sanitization techniques (see Sec. 2.3) reduce the impacts of a range of poisoning attacks. They can be combined with cryptographic techniques for origin and integrity attestation to provide assurances downstream, as recommended in the final report of the National Security Commission on AI [267]. Robust training techniques (see Sec. 2.3) offer different approaches to developing theoretically certified defenses against data poisoning attacks with the intention of providing much-needed information-theoretic guarantees for security. The results are encouraging, but more research is needed to extend this methodology to more general assumptions about data distributions, the ability to handle out-of-distribution inputs, more complex models, multiple data modalities, and better performance. Another challenge is applying these techniques to very large models like LLMs and generative diffusion models, which are becoming targets of attacks [55, 90].

4.2.2. Supply Chain Challenges

The literature on AML shows a trend of designing new attacks that are more difficult to detect. Since the poisoning of AI models can persist through safety training and be triggered by attackers on demand [170], significant concerns arise regarding the potential for models to be created with intentional exploits that are hard for organizations deploying and using models to detect. The potential for attacks against open-source dependencies may be particularly acute in the AI context because organizations and researchers may not be able to audit and identify vulnerabilities encoded into a model's weights in the same way it is often possible to audit open-source software. As users come to rely more on the outputs of AI systems — for example, some research suggests that software engineers who over-rely on AI coding assistants' suggestions may produce less secure code [290, 294, 308] — the potential for malicious actors to subtly manipulate the outputs of AI systems may create increased risk.

Additionally, Goldwasser et al. [142] introduced a new class of attacks: information-theoretically undetectable Trojans that can be planted in ML models. If proven practical, the undetectable nature of such attacks would pose significant challenges for AI supply-chain risk management and increase the importance of preventing insider threat throughout the supply chain. DARPA and NIST have also jointly created TrojAI to research the defense of AI systems from intentional, malicious Trojans by developing the technology to detect and investigate these attacks.

4.2.3. Multimodal Models

MULTIMODAL MODELS have shown great potential for achieving high performance on many ML tasks [27, 30, 258, 304, 435]. However, emerging evidence from practice shows that a redundancy of information across the different modalities does not necessarily make the model more robust against adversarial perturbations of a single modality. Combining modalities and training the model on clean data alone does not seem to improve adversarial robustness. In addition, adversarial training, which is widely used in single modality applications, may become prohibitively expensive as the number of modality combinations increases. Additional effort is required to benefit from the redundant information in order to improve robustness against single modality attacks [417]. Without such an effort, single modality attacks can be effective and compromise multimodal models across a wide range of multimodal tasks despite the information contained in the remaining unperturbed modalities [417, 424]. Moreover, researchers have devised efficient mechanisms for constructing simultaneous attacks on multiple modalities, which suggests that multimodal models might not be more robust against adversarial attacks despite improved performance [77, 333, 415].

> The existence of simultaneous attacks on multimodal models suggests that mitigation techniques that only rely on single modality perturbations are not likely to be robust.

4.2.4. Quantized Models

Quantization is a technique for efficiently deploying models to edge platforms, such as smart phones and IoT devices [138]. It reduces the computational and memory costs of running inference on a given platform by representing the model weights and activations with low-precision data types. For example, quantized models typically use 8-bit integers (int8) or even more compact 4-bit representations instead of the usual 32-bit floating point (float32) numbers for the original non-quantized model.

This technique has been widely used with PredAI and increasingly with GenAI models [108]. However, quantized models inherit the vulnerabilities of the original models and introduce additional weaknesses that make them vulnerable to adversarial attacks. Error amplification from reduced computational precision adversely affects the adversarial robustness of the quantized models. While there are some useful mitigation techniques for PredAI models [216], the effects of quantization on GenAI models have not been studied as thoroughly. Organizations that deploy such models should continuously monitor their behavior. Recent results [118] reveal that widely used quantization methods can be exploited to produce a harmful quantized LLM, even though the full-precision counterpart appears benign, potentially tricking users into deploying the malicious quantized model.

4.2.5. Risk Management in Light of AML

A key question that this taxonomy deliberately leaves aside is how organizations can make decisions about the development and use of AI systems in light of evidence about the increasing diversity of AML attacks and the efficacy and limitations of available mitigations.

Especially in GenAI, some model developers and application builders have moved towards paradigms for testing adversarial risks as part of pre-deployment testing and evaluation of models, such as through a structured process for RED TEAMING [68, 133]. NIST has produced an initial public draft of guidance for model developers on managing risks associated with the misuse of foundation model capabilities, including through pre-deployment evaluations. [275] NIST [273] and Barrett et al. [28] have also developed risk profiles for generative AI systems that map to the NIST AI RMF [274] that may assist model developers and users in assessing risks, including those from adversarial attacks.

However, a persistent challenge remains in the fact that many AML mitigations are empirical in nature and lack theoretical or provable guarantees. In fact, several research results have pointed to theoretical *limits* on AML mitigations, including the impossibility of model-based detection to prevent all impermissible outputs [140] and findings that, so long as a model has any probability of exhibiting an undesired behavior, there exist prompts that can trigger that behavior, implying that any alignment process that attenuates but does not remove an unwanted behavior will remain vulnerable to adversarial prompting attacks. [406]

These theoretical limitations do not obviate the utility of pre-deployment adversarial testing, since such testing can potentially foreclose many attack vectors and thus increase the difficulty of mounting a successful attack above would-be attackers' threshold of effort or capability. However, they suggest that organizations seeking to manage risks related to the development of models with potentially harmful capabilities or the delegation of trust to models in high stakes contexts may need to consider practices and measures beyond adversarial testing to manage the risks associated with AML attacks.

4.2.6. AML and Other AI System Characteristics

A final consideration with respect to adversarial machine learning, and one closely related to questions of risk management, is how to relate and integrate consideration of AML attacks to definitions and processes relating to other desired AI system characteristics.

For example, managing the security of AI systems will require combining mitigations from the field of AML with best practices for the development of secure software from the field of cybersecurity. Understanding and relating these practices to each other, as well as identifying whether there are other key considerations for AI security that fall outside of the scope of either AML or cybersecurity, will be critical as organizations seek to extend existing cybersecurity processes and best practices to address the security of newly adopted AI systems.

Similarly, robustness to AML attacks may play an important role in areas beyond the remit of security, such as in AI safety [275] or in achieving other characteristics of trustworthy AI systems [274]. AML is neither a complete solution to, nor a subset of, any one of these characteristics, and as such, more precisely relating AML attacks and mitigations to processes for achieving these goals and managing risks in AI systems is an area for ongoing work.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM Conference on Computer and Communications Security*, CCS '16, pages 308–318, 2016. https://arxiv.org/abs/1607.00133. doi:10.48550/arXiv.1607.00133.

[2] Mark Abspoel, Daniel Escudero, and Nikolaj Volgushev. Secure training of decision trees with continuous attributes. In *Proceedings on Privacy Enhancing Technologies (PoPETs) 2021, Issue 1*, 2020.

[3] Hojjat Aghakhani, Wei Dai, Andre Manoel, Xavier Fernandes, Anant Kharkar, Christopher Kruegel, Giovanni Vigna, David Evans, Ben Zorn, and Robert Sim. TrojanPuzzle: Covertly poisoning code-suggestion models, 2024. URL: https://arxiv.org/abs/2301.02344, arXiv:2301.02344, doi:10.48550/arXiv.2301.02344.

[4] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In *IEEE European Symposium on Security and Privacy, 2021, Vienna, Austria, September 6-10, 2021*, pages 159–178. IEEE, 2021. URL: https://ieeexplore.ieee.org/document/9581207, doi:10.1109/EuroSP51992.2021.00021.

[5] Meta AI. Llama guard 3 documentation, 2024. Accessed: 2024-08-13. URL: https://llama.meta.com/docs/model-cards-and-prompt-formats/llama-guard-3/.

[6] Meta AI. Prompt guard documentation, 2024. Accessed: 2024-08-13. URL: https://llama.meta.com/docs/model-cards-and-prompt-formats/prompt-guard/.

[7] AI Safety Institute. Systemic ai safety fast grants. https://www.aisi.gov.uk/grants, 2024. Accessed: 2024-08-22.

[8] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine Stochastic Gradient Descent. In *NeurIPS*, 2018. URL: https://arxiv.org/abs/1803.08917, doi:10.48550/arXiv.1803.08917.

[9] Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity, 2023. URL: https://arxiv.org/abs/2308.14132, arXiv:2308.14132, doi:10.48550/arXiv.2308.14132.

[10] Galen Andrew, Peter Kairouz, Sewoong Oh, Alina Oprea, H. Brendan McMahan, and Vinith Suriyakumar. One-shot empirical privacy estimation for federated learning, 2023. URL: https://arxiv.org/abs/2302.03098, arXiv:2302.03098, doi:10.48550/arXiv.2302.03098.

[11] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks, 2024. URL: https://arxiv.org/abs/2404.02151, arXiv:2404.02151, doi:10.48550/arXiv.2404.02151.

[12] Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Eric Winsor, Jerome Wynne, Yarin Gal, and Xander Davies. Agentharm: A benchmark for measuring harmfulness of llm agents, 2024. URL: https://arxiv.org/abs/2410.09024, arXiv:2410.09024.

[13] Anthropic. Model Card and Evaluations for Claude Models. https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf, July 2023. Anthropic.

[14] Anthropic. Anthropic's interactive prompt engineering tutorial. https://github.com/anthropics/prompt-eng-interactive-tutorial, 2024. Accessed: 2024-08-22.

[15] Anthropic. Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet, June 2024. Anthropic.

[16] Anthropic. Expanding our model safety bug bounty program. https://www.anthropic.com/news/model-safety-bug-bounty, 2024. Accessed: 2024-08-22.

[17] Giovanni Apruzzese, Hyrum S Anderson, Savino Dambra, David Freeman, Fabio Pierazzi, and Kevin Roundy. "real attackers don't compute gradients": Bridging the gap between adversarial ml research and practice. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 339–364. IEEE, 2023. URL: https://arxiv.org/abs/2212.14315, doi:10.48550/arXiv.2212.14315.

[18] Arthur. Shield, 2023. URL: https://www.arthur.ai/product/shield.

[19] Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *Int. J. Secur. Netw.*, 10(3):137–150, September 2015. doi:10.1504/IJSN.2015.071829.

[20] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR, 2018. URL: http://proceedings.mlr.press/v80/athalye18a.html, doi:10.48550/arXiv.1802.00420.

[21] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples, 2018. URL: https://arxiv.org/abs/1707.07397, arXiv:1707.07397, doi:10.48550/arXiv.1707.07397.

[22] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR, 26–28 Aug 2020. URL: http://proceedings.mlr.press/v108/bagdasaryan20a.html.

[23] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *AISTATS*. PMLR, 2020. URL: https://proceedings.mlr.press/v108/bagdasaryan20a.html, doi:10.48550/arXiv.1807.00459.

[24] Eugene Bagdasaryan, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. Air gap: Protecting privacy-conscious conversational agents, 2024. URL: https://arxiv.org/abs/2405.05175, arXiv:2405.05175, doi:10.48550/arXiv.2405.05175.

[25] Marieke Bak, Vince Istvan Madai, Marie-Christine Fritzsche, Michaela Th. Mayrhofer, and Stuart McLennan. You can't have ai both ways: Balancing health data privacy and access fairly. *Frontiers in Genetics*, 13, 2022. https://www.frontiersin.org/articles/10.3389/fgene.2022.929453. doi:10.3389/fgene.2022.929453.

[26] Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *NeurIPS 2021 Workshop on Privacy in Machine Learning (PRIML)*, 2021. URL: https://openreview.net/forum?id=Yi2DZTbnBl4, doi:10.48550/arXiv.2201.04845.

[27] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy, 2017. doi:10.48550/ARXIV.1705.09406.

[28] Anthony M. Barrett, Dan Hendrycks, Jessica Newman, and Brandie Nonnecke. *UC Berkeley AI Risk-Management Standards Profile for General-Purpose AI Systems (GPAIS) and Foundation Models*. UC Berkeley Center for Long Term Cybersecurity, 2023. https://cltc.berkeley.edu/seeking-input-and-feedback-ai-risk-management-standards-profile-for-increasingly-multi-purpose-or-general-purpose-ai/. doi:10.48550/ARXIV.2206.08966.

[29] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 515–532, USA, 2019. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/batina.

[30] Khaled Bayoudh, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets. *Vis. Comput.*, 38(8):2939–2970, August 2022. doi:10.1007/s00371-021-02166-7.

[31] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. URL: https://arxiv.org/abs/2303.08112, doi:10.48550/arXiv.2303.08112.

[32] Philipp Benz, Chaoning Zhang, Soomin Ham, Gyusang Karjauv, Adil Cho, and In So Kweon. The triangular trade-off between accuracy, robustness, and fairness. *Workshop on Adversarial Machine Learning in Real-World Computer Vision Systems and Online Challenges (AML-CV) at CVPR*, 2021. URL: https://dl.acm.org/doi/10.1145/3645088, doi:10.1145/3645088.

[33] Jamie Bernardi, Gabriel Mukobi, Hilary Greaves, Lennart Heim, and Markus Anderljung. Societal adaptation to advanced ai, 2024. URL: https://arxiv.org/abs/2405.10295, arXiv:2405.10295, doi:10.48550/arXiv.2405.10295.

[34] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5050–5060.

Curran Associates, Inc., 2019. URL: http://papers.nips.cc/paper/8749-mixmatch-a-holistic-approach-to-semi-supervised-learning.pdf, doi:10.48550/arXiv.2405.10295.

[35] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Model Poisoning Attacks in Federated Learning. In *NeurIPS SECML*, 2018.

[36] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 634–643. PMLR, 09–15 Jun 2019. URL: https://proceedings.mlr.press/v97/bhagoji19a.html, doi:10.48550/arXiv.1811.12470.

[37] Battista Biggio, Igino Corona, Giorgio Fumera, Giorgio Giacinto, and Fabio Roli. Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In *Proceedings of the 10th International Conference on Multiple Classifier Systems*, MCS'11, page 350–359, Berlin, Heidelberg, 2011. Springer-Verlag. URL: https://api.semanticscholar.org/CorpusID:12680508.

[38] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013. doi:10.1007/978-3-642-40994-3_25.

[39] Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In Chun-Nan Hsu and Wee Sun Lee, editors, *Proceedings of the Asian Conference on Machine Learning*, volume 20 of *Proceedings of Machine Learning Research*, pages 97–112, South Garden Hotels and Resorts, Taoyuan, Taiwain, 14–15 Nov 2011. PMLR. URL: https://proceedings.mlr.press/v20/biggio11.html, doi:10.48550/arXiv.2206.00352.

[40] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning, ICML*, 2012. URL: https://arxiv.org/abs/1206.6389, doi:10.48550/arXiv.1206.6389.

[41] Battista Biggio, Konrad Rieck, Davide Ariu, Christian Wressnegger, Igino Corona, Giorgio Giacinto, and Fabio Roli. Poisoning behavioral malware clustering. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, AISec '14, page 27–36, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2666652.2666666.

[42] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, December 2018. URL: https://doi.org/10.1016%2Fj.patcog.2018.07.023, doi:10.1016/j.patcog.2018.07.023.

[43] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NeurIPS*, 2017.

URL: https://papers.nips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b296 39786cb62ef-Paper.pdf.

[44] Rishi Bommasani, Kevin Klyman, Sayash Kapoor, Shayne Longpre, Betty Xiong, Nestor Maslej, and Percy Liang. The foundation model transparency index v1.1: May 2024, 2024. URL: https://arxiv.org/abs/2407.12929, arXiv:2407.12929, doi:10.48550/arXiv.2407.12929.

[45] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 141–159. IEEE, 2021. doi:10.1109/SP40001.2021.00019.

[46] Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. Scaling laws for data poisoning in llms, 2024. URL: https://arxiv.org/abs/2408.02946, arXiv:2408.02946, doi:10.48550/arXiv.2408.02946.

[47] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=SyZI0GWCZ, doi:10.48550/arXiv.1712.04248.

[48] Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 123–132, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451131.

[49] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL: https://arxiv.org/abs/2005.14165, arXiv:2005.14165.

[50] Gon Buzaglo, Niv Haim, Gilad Yehudai, Gal Vardi, and Michal Irani. Reconstructing training data from multiclass neural networks, 2023. URL: https://arxiv.org/abs/2305.03350, arXiv:2305.03350, doi:10.48550/arXiv.2305.03350.

[51] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *NDSS*, 2021. URL: https://arxiv.org/abs/2012.13995, doi:10.48550/arXiv.2012.13995.

[52] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015. URL: https://ieeexplore.ieee.org/document/7163042, doi:10.1109/SP.2015.35.

[53] Nicholas Carlini. Poisoning the unlabeled dataset of Semi-Supervised learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1577–1592. USENIX Association, August 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-poisoning.

[54] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (S&P)*, pages 1519–1519, Los Alamitos, CA, USA, May 2022. IEEE Computer Society. URL: https://doi.ieeecomputersociety.org/10.1109/SP46214.2022.00090, doi:10.1109/SP46214.2022.00090.

[55] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023. URL: https://arxiv.org/abs/2301.13188, arXiv:2301.13188, doi:10.48550/arXiv.2301.13188.

[56] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. https://arxiv.org/abs/2202.07646, 2022. doi:10.48550/ARXIV.2202.07646.

[57] Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149*, 2023. URL: https://arxiv.org/abs/2302.10149, doi:10.48550/arXiv.2302.10149.

[58] Nicholas Carlini, Matthew Jagielski, and Ilya Mironov. Cryptanalytic extraction of neural network models. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 189–218, Cham, 2020. Springer International Publishing. URL: https://arxiv.org/abs/2003.04884, doi:10.48550/arXiv.2003.04884.

[59] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, USENIX '19), pages 267–284, 2019. https://arxiv.org/abs/1802.08232. URL: https://arxiv.org/abs/1802.08232, doi:10.48550/arXiv.1802.08232.

[60] Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Anas Awadalla, Pang Wei Koh, Daphne Ippolito, Katherine Lee, Florian Tramer, et al. Are aligned neural networks adversarially aligned? *arXiv preprint arXiv:2306.15447*, 2023. URL: https://arxiv.org/abs/2306.15447, doi:10.48550/arXiv.2306.15447.

[61] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Itay Yona, Eric Wallace, David Rolnick, and Florian Tramèr. Stealing part of a production language model, 2024. URL: https://arxiv.org/abs/2403.06634, arXiv:2403.06634, doi:10.48550/arXiv.2403.06634.

[62] Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie

Sun, and J. Zico Kolter. (certified!!) adversarial robustness for free!, 2023. URL: https://arxiv.org/abs/2206.10550, arXiv:2206.10550, doi:10.48550/arXiv.2206.10550.

[63] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association, August 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

[64] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 3–14, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3128572.3140444.

[65] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proc. IEEE Security and Privacy Symposium*, 2017. URL: https://arxiv.org/abs/1608.04644, doi:10.48550/arXiv.1608.04644.

[66] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018. URL: https://arxiv.org/abs/1801.01944, doi:10.48550/arXiv.1801.01944.

[67] Stephen Casper, Yuxiao Li, Jiawei Li, Tong Bu, Kevin Zhang, Kaivalya Hariharan, and Dylan Hadfield-Menell. Red teaming deep neural networks with feature synthesis tools. *arXiv preprint arXiv:2302.10894*, 2023. URL: https://arxiv.org/abs/2302.10894, doi:10.48550/arXiv.2302.10894.

[68] Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. Explore, establish, exploit: Red teaming language models from scratch, 2023. URL: https://arxiv.org/abs/2306.09442, arXiv:2306.09442, doi:10.48550/arXiv.2306.09442.

[69] National Cyber Security Center. Introducing our new machine learning security principles, retrieved February 2023 from https://www.ncsc.gov.uk/blog-post/introducing-our-new-machine-learning-security-principles. URL: https://www.ncsc.gov.uk/blog-post/introducing-our-new-machine-learning-security-principles.

[70] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association. URL: https://arxiv.org/abs/1811.02054, doi:10.48550/arXiv.1811.02054.

[71] Hong Chang, Ta Duy Nguyen, Sasi Kumar Murakonda, Ehsan Kazemi, and R. Shokri. On adversarial bias and the robustness of fair machine learning. https://arxiv.org/abs/2006.08669, 2020.

[72] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J.

Pappas, Florian Tramer, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models, 2024. URL: https://arxiv.org/abs/2404.01318, arXiv:2404.01318, doi:10.48550/arXiv.2404.01318.

[73] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023. URL: https://arxiv.org/abs/2310.08419, doi:10.48550/arXiv.2310.08419.

[74] Harsh Chaudhari, John Abascal, Alina Oprea, Matthew Jagielski, Florian Tramèr, and Jonathan Ullman. SNAP: Efficient extraction of private properties with poisoning. In *2023 IEEE Symposium on Security and Privacy (S&P)*, 2023. URL: https://arxiv.org/abs/2208.12348, doi:10.48550/arXiv.2208.12348.

[75] Harsh Chaudhari, Giorgio Severi, John Abascal, Matthew Jagielski, Christopher A. Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, and Alina Oprea. Phantom: General trigger attacks on retrieval augmented language generation, 2024. URL: https://arxiv.org/abs/2405.20485, arXiv:2405.20485, doi:10.48550/arXiv.2405.20485.

[76] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. https://arxiv.org/abs/1811.03728, 2018. URL: https://arxiv.org/abs/1811.03728, doi:10.48550/arXiv.1811.03728.

[77] Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. https://arxiv.org/abs/1712.02051, 2017. URL: https://arxiv.org/abs/1712.02051, doi:10.48550/ARXIV.1712.02051.

[78] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4658–4664. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi:10.24963/ijcai.2019/647.

[79] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. HopSkipJumpAttack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 1277–1294. IEEE, 2020. doi:10.1109/SP40000.2020.00045.

[80] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 15–26, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3128572.3140448.

[81] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. *ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector*, page 52–68. Springer International Publishing, 2019. URL: http://dx.doi.org/10.1007/978-3-030-10925-7_4, doi:10.1007/978-3-030-10925-7_4.

[82] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, ACSAC '21, page 554–569, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3485832.3485837.

[83] Xiaoyi Chen, Siyuan Tang, Rui Zhu, Shijun Yan, Lei Jin, Zihao Wang, Liya Su, Zhikun Zhang, XiaoFeng Wang, and Haixu Tang. The Janus interface: How fine-tuning in large language models amplifies the privacy risks, 2024. URL: https://arxiv.org/abs/2310.15469, arXiv:2310.15469, doi:10.48550/arXiv.2310.15469.

[84] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. URL: https://arxiv.org/abs/1712.05526, doi:10.48550/arXiv.1712.05526.

[85] Heng-Tze Cheng and Romal Thoppilan. LaMDA: Towards Safe, Grounded, and High-Quality Dialog Models for Everything. https://ai.googleblog.com/2022/01/lamda-towards-safe-grounded-and-high.html, 2022. Google Brain. URL: https://research.google/blog/lamda-towards-safe-grounded-and-high-quality-dialog-models-for-everything/.

[86] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=rJlk6iRqKX, doi:10.48550/arXiv.1807.04457.

[87] Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2020. URL: https://openreview.net/forum?id=SklTQCNtvS, doi:10.48550/arXiv.1909.10773.

[88] Alesia Chernikova and Alina Oprea. FENCE: Feasible evasion attacks on neural networks in constrained environments. *ACM Transactions on Privacy and Security (TOPS) Journal*, 2022. URL: https://arxiv.org/abs/1909.10480, doi:10.48550/arXiv.1909.10480.

[89] Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1964–1974. PMLR, 18–24 Jul 2021. URL: https://proceedings.mlr.press/v139/choquette-choo21a.html, doi:10.48550/arXiv.2007.14321.

[90] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? https://arxiv.org/abs/2212.05400, 2022. doi:10.48550/ARXIV.2212.05400.

[91] Antonio Emanuele Cinà, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard A. Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. Wild patterns reloaded: A survey of machine learning security

against training data poisoning. *ACM Computing Surveys*, March 2023. URL: https://doi.org/10.1145%2F3585385, doi:10.1145/3585385.

[92] Jack Clark and Raymond Perrault. 2022 AI index report. https://aiindex.stanford.edu/wp-content/uploads/2022/03/2022-AI-Index-Report_Master.pdf, 2022. Human Centered AI, Stanford University.

[93] Joseph Clements, Yuzhe Yang, Ankur Sharma, Hongxin Hu, and Yingjie Lao. Rallying adversarial techniques against deep learning for network security, 2019. URL: https://arxiv.org/abs/1903.11688, doi:10.48550/ARXIV.1903.11688.

[94] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019. URL: https://proceedings.mlr.press/v97/cohen19c.html.

[95] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[96] Gabriela F. Cretu, Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo, and Angelos D. Keromytis. Casting out demons: Sanitizing training data for anomaly sensors. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 81–95, 2008. URL: https://ieeexplore.ieee.org/document/4531146, doi:10.1109/SP.2008.11.

[97] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL: https://openreview.net/forum?id=SSKZPJCt7B, doi:10.48550/arXiv.2010.09670.

[98] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’04, page 99–108, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/1014052.1014066.

[99] DARPA. DARPA AI Cyber Challenge Aims to Secure Nation’s Most Critical Software, 2023. Accessed: 2024-08-22. URL: https://www.darpa.mil/news-events/2023-08-09.

[100] Emiliano De Cristofaro. A critical overview of privacy in machine learning. *IEEE Security & Privacy*, 19(4):19–27, 2021. doi:10.1109/MSEC.2021.3076443.

[101] Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate attacks and defenses for LLM agents, 2024. URL: https://arxiv.org/abs/2406.13352, arXiv:2406.13352, doi:10.48550/arXiv.2406.13352.

[102] DeepMind. Building safer dialogue agents. https://www.deepmind.com/blog/building-safer-dialogue-agents, 2022. Online.

[103] Luca Demetrio, Battista Biggio, Giovanni Lagorio, Fabio Roli, and Alessandro Armando. Functionality-preserving black-box optimization of adversarial windows malware. *IEEE Transactions on Information Forensics and Security*, 16:3469–3478, 2021. URL: https://ieeexplore.ieee.org/document/9437194, doi:10.1109/TIFS.2021.3082330.

[104] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 321–338. USENIX Association, 2019. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/demontis.

[105] Serguei Denissov, Hugh Brendan McMahan, J Keith Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for SGD via optimal private linear operators on adaptive streams. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL: https://openreview.net/forum?id=i9XrHJoyLqJ, doi:10.48550/arXiv.2202.08312.

[106] Leon Derczynski. Garak: LLM vulnerability scanner. https://github.com/leondz/garak, 2024. Accessed: 2024-08-18.

[107] Leon Derczynski, Erick Galinkin, Jeffrey Martin, Subho Majumdar, and Nanna Inie. garak: A framework for security probing large language models, 2024. URL: https://arxiv.org/abs/2406.11036, arXiv:2406.11036, doi:10.48550/arXiv.2406.11036.

[108] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=OUIFPHEgJU, doi:10.48550/arXiv.2305.14314.

[109] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606. PMLR, 2019. URL: https://arxiv.org/abs/1803.02815, doi:10.48550/arXiv.1803.02815.

[110] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM Symposium on Principles of Database Systems*, PODS '03, pages 202–210. ACM, 2003. URL: https://crypto.stanford.edu/seclab/sem-03-04/psd.pdf.

[111] Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush R. Varshney. Is there a trade-off between fairness and accuracy? A perspective using mismatched hypothesis testing. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. URL: https://arxiv.org/abs/1910.07870, doi:10.48550/arXiv.1910.07870.

[112] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceed-*

ings, Part II, pages 1–12, 2006. URL: http://dx.doi.org/10.1007/11787006_1, doi:10.1007/11787006_1.

[113] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Conference on Theory of Cryptography*, TCC '06, pages 265–284, New York, NY, USA, 2006.

[114] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! A survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017. URL: https://privacytools.seas.harvard.edu/publications/exposed-survey-attacks-private-data.

[115] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *IEEE Symposium on Foundations of Computer Science*, FOCS '15, 2015. URL: https://privacytools.seas.harvard.edu/files/privacytools/files/robust.pdf.

[116] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *Annual International Cryptology Conference*, pages 469–480. Springer, 2008. URL: https://link.springer.com/chapter/10.1007/978-3-540-85174-5_26, doi:0.1007/978-3-540-85174-5_26.

[117] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*, 2017. URL: https://arxiv.org/abs/1712.06751, doi:10.48550/arXiv.1712.06751.

[118] Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting LLM Quantization, 2024. URL: https://arxiv.org/abs/2405.18137, arXiv:2405.18137, doi:10.48550/arXiv.2405.18137.

[119] Gemini Team et al. Gemini: A family of highly capable multimodal models. https://arxiv.org/abs/2312.11805, 2023. arXiv:2312.11805.

[120] ETSI Group Report SAI 005. Securing artificial intelligence (SAI); mitigation strategy report, retrieved February 2023 from https://www.etsi.org/deliver/etsi_gr/SAI/001_099/005/01.01.01_60/gr_SAI005v010101p.pdf. URL: https://www.etsi.org/deliver/etsi_gr/SAI/001_099/005/01.01.01_60/gr_SAI005v010101p.pdf.

[121] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. URL: https://ieeexplore.ieee.org/document/8578273, doi:10.1109/CVPR.2018.00175.

[122] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1625–1634. Computer Vision Foundation / IEEE Computer Society, 2018. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Eykholt_Robust_Physical-World_Attacks_CVPR_2018_paper.html, doi:10.1109/CVPR.2018.00175.

[123] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local Model Poi-

soning Attacks to Byzantine-Robust Federated Learning. In *USENIX Security*, 2020.

[124] Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. online: https://arxiv.org/abs/2210.14707, 2022. doi:10.48550/ARXIV.2210.14707.

[125] Georgios Fatouros, John Soldatos, Kalliopi Kouroumali, Georgios Makridis, and Dimosthenis Kyriazis. Transforming sentiment analysis in the financial domain with chatgpt. *Machine Learning with Applications*, 14:100508, 2023. URL: https://www.sciencedirect.com/science/article/pii/S2666827023000610, doi:10.1016/j.mlwa.2023.100508.

[126] Vitaly Feldman. Does learning require memorization? A short tale about a long tail. In *ACM Symposium on Theory of Computing*, STOC '20, pages 954–959, 2020. https://arxiv.org/abs/1906.05271.

[127] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. URL: https://arxiv.org/abs/2008.03703, doi:10.48550/arXiv.2008.03703.

[128] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: Generating training time adversarial data with auto-encoder. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/1ce83e5d4135b07c0b82afffbe2b3436-Paper.pdf, doi:10.48550/arXiv.1905.09027.

[129] Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release, 2021. URL: https://arxiv.org/abs/2103.02683, doi:10.48550/ARXIV.2103.02683.

[130] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2810103.2813677.

[131] Aymeric Fromherz, Klas Leino, Matt Fredrikson, Bryan Parno, and Corina Pasareanu. Fast geometric projections for local robustness certification. In *International Conference on Learning Representations*, 2021. URL: https://openreview.net/forum?id=zWy1uxjDdZJ, doi:10.48550/arXiv.2002.04742.

[132] Pranav Gade, Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. Badllama: cheaply removing safety fine-tuning from llama 2-chat 13b, 2024. URL: https://arxiv.org/abs/2311.00117, arXiv:2311.00117, doi:10.48550/arXiv.2311.00117.

[133] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones,

Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned, 2022. URL: https://arxiv.org/abs/2209.07858, arXiv:2209.07858, doi:10.48550/arXiv.2209.07858.

[134] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 619–633, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3243734.3243834.

[135] Simson Garfinkel, John Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Communications of the ACM*, 62:46–53, 02 2019. URL: https://dl.acm.org/doi/10.1145/3287287, doi:10.1145/3287287.

[136] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (S&P)*, pages 3–18, 2018. URL: https://ieeexplore.ieee.org/document/8418593, doi:10.1109/SP.2018.00058.

[137] Jonas Geiping, Liam H Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations*, 2021. URL: https://openreview.net/forum?id=01olnfLIbD, doi:10.48550/arXiv.2009.02276.

[138] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference, 2021. URL: https://arxiv.org/abs/2103.13630, arXiv:2103.13630, doi:10.48550/arXiv.2103.13630.

[139] Antonio A. Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making ai forget you: Data deletion in machine learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc. URL: https://arxiv.org/abs/1907.05012, doi:10.48550/arXiv.1907.05012.

[140] David Glukhov, Ilia Shumailov, Yarin Gal, Nicolas Papernot, and Vardan Papyan. LLM censorship: A machine learning challenge or a computer security problem?, 2023. URL: https://arxiv.org/abs/2307.10719, arXiv:2307.10719, doi:10.48550/arXiv.2307.10719.

[141] Micah Goldblum, Avi Schwarzschild, Ankit Patel, and Tom Goldstein. Adversarial attacks on machine learning systems for high-frequency trading. In *Proceedings of the Second ACM International Conference on AI in Finance*, ICAIF '21, New York, NY,

USA, 2021. Association for Computing Machinery. doi:10.1145/3490354.3494367.

[142] Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models. https://arxiv.org/abs/2204.06974, 2022. arXiv. doi:10.48550/ARXIV.2204.06974.

[143] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[144] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL: http://arxiv.org/abs/1412.6572, doi:10.48550/arXiv.1412.6572.

[145] Kai Greshake. Prompt injection defenses should suck less. https://kai-greshake.de/posts/approaches-to-pi-defense/, March 2024. Accessed: 2024-08-22.

[146] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. *arXiv preprint arXiv:2302.12173*, 2023. URL: https://arxiv.org/abs/2302.12173, doi:10.48550/arXiv.2302.12173.

[147] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in neural information processing systems*, 26, 2013. URL: https://papers.nips.cc/paper_files/paper/2013/hash/e034fb6b66aacc1d48f445ddfb08da98-Abstract.html.

[148] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. URL: https://ieeexplore.ieee.org/document/8685687, doi:10.1109/ACCESS.2019.2909068.

[149] Rachid Guerraoui, Arsany Guirguis, Jérémy Plassmann, Anton Ragot, and Sébastien Rouault. Garfield: System support for byzantine machine learning (regular paper). In *DSN*. IEEE, 2021. URL: https://arxiv.org/abs/2010.05888, doi:10.48550/arXiv.2010.05888.

[150] Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL: https://aclanthology.org/2021.emnlp-main.464, doi:10.18653/v1/2021.emnlp-main.464.

[151] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges, 2024. URL: https://arxiv.org/abs/2402.01680, arXiv:2402.01680, doi:10.48550/arXiv.2402.01680.

[152] Niv Haim, Gal Vardi, Gilad Yehudai, michal Irani, and Ohad Shamir. Reconstructing training data from trained neural networks. In Alice H. Oh, Alekh Agarwal, Danielle

Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL: https://openreview.net/forum?id=Sxk8Bse3RKO, doi:10.48550/arXiv.2206.07758.

[153] Danny Halawi, Alexander Wei, Eric Wallace, Tony T. Wang, Nika Haghtalab, and Jacob Steinhardt. Covert malicious finetuning: Challenges in safeguarding llm adaptation, 2024. URL: https://arxiv.org/abs/2406.20053, arXiv:2406.20053, doi:10.48550/arXiv.2406.20053.

[154] Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs, 2024. URL: https://arxiv.org/abs/2406.18495, arXiv:2406.18495, doi:10.48550/arXiv.2406.18495.

[155] Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. LLM multi-agent systems: Challenges and open problems, 2024. URL: https://arxiv.org/abs/2402.03578, arXiv:2402.03578, doi:10.48550/arXiv.2402.03578.

[156] Drew Harwell. ID.me gathers lots of data besides face scans, including locations. Scammers still have found a way around it., retrieved December 2024. URL: https://www.washingtonpost.com/technology/2022/02/11/idme-facial-recognition-fraud-scams-irs/.

[157] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. SPECTRE: Defending against backdoor attacks using robust statistics. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4129–4139. PMLR, 18–24 Jul 2021. URL: https://proceedings.mlr.press/v139/hayase21a.html, doi:10.48550/arXiv.2104.11315.

[158] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety, 2022. URL: https://arxiv.org/abs/2109.13916, arXiv:2109.13916, doi:10.48550/arXiv.2109.13916.

[159] Isaac Hepworth, Kara Olive, Kingshuk Dasgupta, Michael Le, Mark Lodato, Mihai Maruseac, Sarah Meiklejohn, Shamik Chaudhuri, and Tehila Minkus. Securing the ai software supply chain. Technical report, Google, 2024. URL: https://research.google/pubs/securing-the-ai-software-supply-chain/.

[160] Keegan Hines, Gary Lopez, Matthew Hall, Federico Zarfati, Yonatan Zunger, and Emre Kiciman. Defending against indirect prompt injection attacks with spotlighting, 2024. URL: https://arxiv.org/abs/2403.14720, arXiv:2403.14720, doi:10.48550/arXiv.2403.14720.

[161] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL: https://arxiv.org/abs/2203.15556, arXiv:2203.15556.

[162] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008. URL: https://pubmed.ncbi.nlm.nih.gov/18769715/, doi:10.1371/journal.pgen.1000167.

[163] Xiaoling Hu, Xiao Lin, Michael Cogswell, Yi Yao, Susmit Jha, and Chao Chen. Trigger hunting with a topological prior for trojan detection. In *International Conference on Learning Representations*, 2022. URL: https://openreview.net/forum?id=TXsjU8BaibT, doi:10.48550/arXiv.2110.08335.

[164] Zhengmian Hu, Gang Wu, Saayan Mitra, Ruiyi Zhang, Tong Sun, Heng Huang, and Viswanathan Swaminathan. Token-level adversarial prompt detection based on perplexity measures and contextual information, 2024. URL: https://arxiv.org/abs/2311.11509, arXiv:2311.11509, doi:10.48550/arXiv.2311.11509.

[165] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? *arXiv preprint arXiv:2205.12628*, 2022. URL: https://arxiv.org/abs/2205.12628, doi:10.48550/arXiv.2205.12628.

[166] W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12080–12091. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper/2020/file/8ce6fc704072e351679ac97d4a985574-Paper.pdf, doi:10.48550/arXiv.2004.00225.

[167] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents, 2022. URL: https://arxiv.org/abs/2201.07207, arXiv:2201.07207, doi:10.48550/arXiv.2201.07207.

[168] Xijie Huang, Moustafa Alzantot, and Mani Srivastava. NeuronInspect: Detecting backdoors in neural networks via output explanations, 2019. URL: https://arxiv.org/abs/1911.07399, doi:10.48550/ARXIV.1911.07399.

[169] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Yang Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong

Chen, and Yue Zhao. Position: TrustLLM: Trustworthiness in large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL: https://openreview.net/forum?id=bWUU0LwwMp, doi:10.48550/arXiv.2401.05561.

[170] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. Sleeper agents: Training deceptive llms that persist through safety training, 2024. URL: https://arxiv.org/abs/2401.05566, arXiv:2401.05566, doi:10.48550/arXiv.2201.07207.

[171] W. Nicholson Price II. Risks and remedies for artificial intelligence in health care. https://www.brookings.edu/research/risks-and-remedies-for-artificial-intelligence-in-health-care/, 2019. Brookings Report.

[172] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2142–2151. PMLR, 2018. URL: http://proceedings.mlr.press/v80/ilyas18a.html, doi:10.48550/arXiv.1804.08598.

[173] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019. URL: https://openreview.net/forum?id=BkMiWhR5K7, doi:10.48550/arXiv.1807.07978.

[174] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf, doi:10.48550/arXiv.1905.02175.

[175] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2008–2016. PMLR, 2021. URL: http://proceedings.mlr.press/v130/izzo21a.html, doi:10.48550/arXiv.2002.10077.

[176] Shahin Jabbari, Han-Ching Ou, Himabindu Lakkaraju, and Milind Tambe. An empirical study of the trade-offs between interpretability and fairness. In *ICML Workshop on Human Interpretability in Machine Learning, International Conference on Machine Learning (ICML)*, 2020. URL: https://teamcore.seas.harvard.edu/files/team

core/files/2020_jabbari_paper_32.pdf.

[177] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association. URL: https://arxiv.org/abs/1909.01838, doi:10.48550/arXiv.1909.01838.

[178] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi Malvajerdi, and Jonathan Ullman. Differentially private fair learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 3000–3008. PMLR, 2019. URL: https://arxiv.org/abs/1812.02696, doi:10.48550/arXiv.1812.02696.

[179] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *IEEE Symposium on Security and Privacy (S&P)*, pages 19–35, 2018. URL: https://arxiv.org/abs/1804.00308, doi:10.48550/arXiv.1804.00308.

[180] Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. Subpopulation data poisoning attacks. In *Proceedings of the ACM Conference on Computer and Communications Security*, CCS, 2021. URL: https://arxiv.org/abs/2006.14026, doi:10.48550/arXiv.2006.14026.

[181] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private SGD? In *Advances in Neural Information Processing Systems*, volume 33, pages 22205–22216, 2020. URL: https://proceedings.neurips.cc/paper/2020/file/fc4ddc15f9f4b4b06ef7844d6bb53abf-Paper.pdf, doi:10.48550/arXiv.2006.07709.

[182] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023. URL: https://arxiv.org/abs/2309.00614, arXiv:2309.00614, doi:10.48550/arXiv.2309.00614.

[183] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 1895–1912, USA, 2019. USENIX Association. URL: https://arxiv.org/abs/1902.08874, doi:10.48550/arXiv.1902.08874.

[184] Bargav Jayaraman and David Evans. Are attribute inference attacks just imputation? In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1569–1582, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3548606.3560663.

[185] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September

2017. Association for Computational Linguistics. URL: https://aclanthology.org/D17-1215, doi:10.18653/v1/D17-1215.

[186] Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. Artprompt: Ascii art-based jailbreak attacks against aligned llms, 2024. URL: https://arxiv.org/abs/2402.11753, arXiv:2402.11753, doi:10.48550/arXiv.2402.11753.

[187] Pengfei Jing, Qiyi Tang, Yuefeng Du, Lei Xue, Xiapu Luo, Ting Wang, Sen Nie, and Shi Wu. Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3237–3254. USENIX Association, August 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/jing.

[188] Nikola Jovanovic, Robin Staab, and Martin Vechev. Watermark Stealing in Large Language Models. In *Proceedings of the 41-st International Conference on Machine Learning*, PMLR 235, June 2024. URL: https://files.sri.inf.ethz.ch/website/papers/jovanovic2024watermarkstealing.pdf, doi:10.48550/arXiv.2402.19361.

[189] Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5213–5225. PMLR, 18–24 Jul 2021. URL: https://proceedings.mlr.press/v139/kairouz21b.html.

[190] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2019. URL: https://arxiv.org/abs/1912.04977, doi:10.48550/ARXIV.1912.04977.

[191] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In Rupak Majumdar and Viktor Kunčak, editors, *Computer Aided Verification*, pages 97–117, Cham, 2017. Springer International Publishing. URL: https://arxiv.org/abs/1702.01135, doi:10.48550/arXiv.1702.01135.

[192] Michael Kearns and Ming Li. Learning in the presence of malicious errors. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 267–280, New York, NY, USA, 1988. Association for Computing Machinery.

doi:10.1145/62212.62238.

[193] Alaa Khaddaj, Guillaume Leclerc, Aleksandar Makelov, Kristian Georgiev, Hadi Salman, Andrew Ilyas, and Aleksander Madry. Rethinking backdoor attacks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16216–16236. PMLR, 23–29 Jul 2023. URL: https://proceedings.mlr.press/v202/khaddaj23a.html, doi:10.48550/arXiv.2307.10163.

[194] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models, 2023. URL: https://arxiv.org/abs/2301.10226, arXiv:2301.10226, doi:10.48550/arXiv.2301.10226.

[195] Marius Kloft and Pavel Laskov. Security analysis of online centroid anomaly detection. *Journal of Machine Learning Research*, 13(118):3681–3724, 2012. URL: http://jmlr.org/papers/v13/kloft12b.html.

[196] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017. URL: https://arxiv.org/abs/1703.04730, doi:10.48550/arXiv.1703.04730.

[197] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences, 2023. URL: https://arxiv.org/abs/2302.08582, arXiv:2302.08582, doi:10.48550/arXiv.2302.08582.

[198] Moshe Kravchik, Battista Biggio, and Asaf Shabtai. Poisoning attacks on cyber attack detectors for industrial control systems. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 116–125, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3412841.3441892.

[199] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning – industry perspectives. https://arxiv.org/abs/2002.05646, 2020. doi:10.48550/ARXIV.2002.05646.

[200] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. https://arxiv.org/abs/1607.02533, 2016. doi:10.48550/ARXIV.1607.02533.

[201] Keita Kurita and Paul Michel amd Graham Neubig. Weight poisoning attacks on pretrained models, 2020. URL: https://arxiv.org/abs/2004.06660, arXiv:2004.06660, doi:10.48550/arXiv.2004.06660.

[202] E. La Malfa and M. Kwiatkowska. The king is naked: On the notion of robustness for natural language processing. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, volume 10, page 11047–57. Association for the Advancement of Artificial Intelligence, 2022. URL: https://arxiv.org/abs/2112.07605, doi:10.48550/arXiv.2112.07605.

[203] Ricky Laishram and Vir Virander Phoha. Curie: A method for protecting SVM classi-

fier from poisoning attack. *CoRR*, abs/1606.01584, 2016. URL: http://arxiv.org/abs/1606.01584, arXiv:1606.01584, doi:10.48550/arXiv.1606.01584.

[204] Lakera. Guard, 2023. URL: https://www.lakera.ai/.

[205] Harry Langford, Ilia Shumailov, Yiren Zhao, Robert D. Mullins, and Nicolas Papernot. Architectural neural backdoors from first principles. *CoRR*, abs/2402.06957, 2024. URL: https://doi.org/10.48550/arXiv.2402.06957, arXiv:2402.06957, doi:10.48550/ARXIV.2402.06957.

[206] Learn Prompting. Defensive measures, 2023. URL: https://learnprompting.org/docs/category/-defensive-measures.

[207] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672. IEEE, 2019. doi:10.1109/SP.2019.00044.

[208] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association. URL: https://arxiv.org/abs/1906.11798, doi:10.48550/arXiv.1906.11798.

[209] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *International Conference on Learning Representations*, 2021. URL: https://openreview.net/forum?id=YUGG2tFuPM, doi:10.48550/arXiv.2006.14768.

[210] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. arXiv:2005.11401.

[211] Linyi Li, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. In *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, 22-26 May 2023*. IEEE, 2023. URL: https://arxiv.org/abs/2009.04131, doi:10.48550/arXiv.2009.04131.

[212] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024. URL: https://arxiv.org/abs/2403.03218, arXiv:2403.03218, doi:

10.48550/arXiv.2403.03218.

[213] Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. Hidden backdoors in human-centric language models. In Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi, editors, *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 3123–3140. ACM, 2021. doi:10.1145/3460120.3484576.

[214] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18:2088–2105, 2021. URL: https://arxiv.org/abs/1909.02742, doi:10.48550/arXiv.1909.02742.

[215] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy-Chowdhury, and Ananthram Swami. Adversarial perturbations against real-time video classification systems. *CoRR*, abs/1807.00458, 2018. URL: http://arxiv.org/abs/1807.00458, arXiv:1807.00458, doi:10.14722/ndss.2019.23202.

[216] Ji Lin, Chuang Gan, and Song Han. Defensive quantization: When efficiency meets robustness. *ArXiv*, abs/1904.08444, 2019. URL: https://api.semanticscholar.org/CorpusID:53502621, doi:10.48550/arXiv.1904.08444.

[217] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In Michael Bailey, Sotiris Ioannidis, Manolis Stamatogiannakis, and Thorsten Holz, editors, *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Proceedings*, Lecture Notes in Computer Science, pages 273–294. Springer Verlag, 2018. URL: https://link.springer.com/chapter/10.1007/978-3-030-00470-5_13, doi:10.1007/978-3-030-00470-5_13.

[218] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017. URL: https://openreview.net/forum?id=Sys6GJqxl, doi:10.48550/arXiv.1611.02770.

[219] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against LLM-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023. URL: https://arxiv.org/abs/2306.05499, doi:10.48550/arXiv.2306.05499.

[220] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking ChatGPT via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023. URL: https://arxiv.org/abs/2305.13860, doi:10.48550/arXiv.2305.13860.

[221] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. ABS: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 1265–1282, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3319535.3363216.

[222] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*. The Internet Society, 2018. URL: http://dblp.uni-trier.de/db/conf/ndss/ndss2018.html#LiuMALZW018.

[223] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 182–199, Cham, 2020. Springer International Publishing. URL: https://arxiv.org/abs/2007.02343, doi:10.48550/arXiv.2007.02343.

[224] Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, and toxicity, 2023. URL: https://arxiv.org/abs/2305.13169, arXiv:2305.13169, doi:10.48550/arXiv.2305.13169.

[225] Martin Bertran Lopez, Shuai Tang, Michael Kearns, Jamie Morgenstern, Aaron Roth, and Zhiwei Steven Wu. Scalable membership inference attacks via quantile regression. In *NeurIPS 2023*, 2023. URL: https://www.amazon.science/publications/scalable-membership-inference-attacks-via-quantile-regression.

[226] Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 641–647, New York, NY, USA, 2005. Association for Computing Machinery. doi:10.1145/1081870.1081950.

[227] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles, 2017. URL: https://arxiv.org/abs/1707.03501, arXiv:1707.03501, doi:10.48550/arXiv.1707.03501.

[228] Yiwei Lu, Gautam Kamath, and Yaoliang Yu. Indiscriminate data poisoning attacks on neural networks. https://arxiv.org/abs/2204.09092, 2022. doi:10.48550/ARXIV.2204.09092.

[229] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE Computer Society, 2023. URL: https://arxiv.org/abs/2302.00539, doi:10.48550/arXiv.2302.00539.

[230] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[231] Pooria Madani and Natalija Vlajic. Robustness of deep autoencoder in intrusion detection under adversarial contamination. In *HoTSoS '18: Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security*, pages 1–8, 04 2018. doi:10.1145/3190619.3190637.

[232] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In

6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJzIBfZAb, doi:10.48550/arXiv.1706.06083.

[233] Saeed Mahloujifar, Esha Ghosh, and Melissa Chase. Property inference from poisoning. In *2022 IEEE Symposium on Security and Privacy (S&P)*, pages 1120–1137, 2022. URL: https://ieeexplore.ieee.org/document/9833623, doi:10.1109/SP46214.2022.9833623.

[234] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for LLMs, 2024. URL: https://arxiv.org/abs/2401.06121, arXiv:2401.06121, doi:10.48550/arXiv.2401.06121.

[235] Neal Mangaokar, Ashish Hooda, Jihye Choi, Shreyas Chandrashekaran, Kassem Fawaz, Somesh Jha, and Atul Prakash. Prp: Propagating universal perturbations to attack large language model guard-rails, 2024. URL: https://arxiv.org/abs/2402.15911, arXiv:2402.15911, doi:10.48550/arXiv.2402.15911.

[236] James Manyika and Sissie Hsiao. An overview of Bard: an early experiment with generative AI. https://ai.google/static/documents/google-about-bard.pdf, February 2023. Google.

[237] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal, 2024. URL: https://arxiv.org/abs/2402.04249, arXiv:2402.04249, doi:10.48550/arXiv.2402.15911.

[238] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Las Vegas, NV, USA, 2007. URL: https://ieeexplore.ieee.org/document/4389483, doi:10.1109/FOCS.2007.66.

[239] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. *arXiv preprint arXiv:2312.02119*, 2023. URL: https://arxiv.org/abs/2312.02119, doi:10.48550/arXiv.2312.02119.

[240] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 691–706. IEEE, 2019. doi:10.1109/SP.2019.00029.

[241] Melissa Heikkilä. This new data poisoning tool lets artists fight back against generative AI. https://www.technologyreview.com/2023/10/23/1082189/data-poisoning-artists-fight-generative-ai/, October 2023. MIT Technology Review.

[242] El Mahdi El Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). In *NeurIPS*, 2021. URL: https://arxiv.org/abs/2008.00742, doi:10.48550/arXiv.2

008.00742.

[243] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The Hidden Vulnerability of Distributed Learning in Byzantium. In *ICML*, 2018. URL: https://arxiv.org/abs/1802.07927, doi:10.48550/arXiv.1802.07927.

[244] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *ICLR*, 2021. URL: https://arxiv.org/abs/2003.00010, doi:10.48550/arXiv.2003.00010.

[245] Dang Minh, H. Xiang Wang, Y. Fen Li, and Tan N. Nguyen. You can't have AI both ways: Balancing health data privacy and access fairly. *Artificial Intelligence Review volume*, 55:3503–3568, 2022. https://doi.org/10.1007/s10462-021-10088-y. doi:10.3389/fgene.2022.929453.

[246] Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019. URL: https://arxiv.org/abs/1908.10530, doi:10.48550/arXiv.1908.10530.

[247] Margaret Mitchell, Giada Pistilli, Yacine Jernite, Ezinwanne Ozoani, Marissa Gerchick, Nazneen Rajani, Sasha Luccioni, Irene Solaiman, Maraim Masoud, Somaieh Nikpoor, Carlos Muñoz Ferrandis, Stas Bekman, Christopher Akiki, Danish Contractor, David Lansky, Angelina McMillan-Major, Tristan Thrush, Suzana Ilić, Gérard Dupont, Shayne Longpre, Manan Dey, Stella Biderman, Douwe Kiela, Emi Baylor, Teven Le Scao, Aaron Gokaslan, Julien Launay, and Niklas Muennighoff. BigScience Large Open-science Open-access Multilingual Language Model. https://huggingface.co/bigscience/bloom, 2022. Hugging Face.

[248] MITRE. ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems, retrieved December 2024. URL: https://atlas.mitre.org/.

[249] MITRE ATLAS. AML.M0001: Limit Model Artifact Release. https://atlas.mitre.org/mitigations/AML.M0001, 2023. Last Modified: 12 October 2023.

[250] MITRE ATLAS. AML.M0002: Passive ML Output Obfuscation. https://atlas.mitre.org/mitigations/AML.M0002, 2023. Last Modified: 12 October 2023.

[251] MITRE ATLAS. AML.M0004: Restrict Number of ML Model Queries. https://atlas.mitre.org/mitigations/AML.M0004, 2023. Last Modified: 12 October 2023.

[252] MITRE ATLAS. AML.M0000: Limit Release of Public Information. https://atlas.mitre.org/mitigations/AML.M0000, 2024. Last Modified: 12 January 2024.

[253] Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38, 2017. doi:10.1109/SP.2017.12.

[254] Seungyong Moon, Gaon An, and Hyun Oh Song. Parsimonious black-box adversarial attacks via efficient combinatorial optimization. In *International Conference on Machine Learning (ICML)*, 2019. URL: https://arxiv.org/abs/1905.06635, doi:10.48550/arXiv.1905.06635.

[255] Olivia Moore. How Are Consumers Using Generative AI? *Andreessen Horowitz (a16z)*, 2023. URL: https://a16z.com/how-are-consumers-using-generative-ai/.

[256] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard.

Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. URL: https://arxiv.org/abs/1610.08401, doi:10.48550/arXiv.1610.08401.
[257] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A simple and accurate method to fool deep neural networks. https://arxiv.org/abs/1511.04599, 2015. doi:10.48550/ARXIV.1511.04599.
[258] Ghulam Muhammad, Fatima Alshehri, Fakhri Karray, Abdulmotaleb El Saddik, Mansour Alsulaiman, and Tiago H. Falk. A comprehensive survey on multimodal medical signals fusion for smart healthcare systems. *Information Fusion*, 76:355–375, 2021. URL: https://www.sciencedirect.com/science/article/pii/S1566253521001330, doi:10.1016/j.inffus.2021.06.007.
[259] Sasi Kumar Murakonda and Reza Shokri. ML Privacy Meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning, 2020. URL: https://arxiv.org/abs/2007.09339, doi:10.48550/ARXIV.2007.09339.
[260] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, 2017. URL: https://arxiv.org/abs/1708.08689, doi:10.48550/arXiv.1708.08689.
[261] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL: https://arxiv.org/abs/2112.09332, arXiv:2112.09332, doi:10.48550/arXiv.2112.09332.
[262] Nina Narodytska and Shiva Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1310–1318, 2017. URL: https://ieeexplore.ieee.org/document/8014906, doi:10.1109/CVPRW.2017.172.
[263] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pages 1631–1648. USENIX Association, 2023. URL: https://www.usenix.org/conference/usenixsecurity23/presentation/nasr, doi:10.48550/arXiv.2302.07956.
[264] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, pages 739–753. IEEE, 2019. URL: https://arxiv.org/abs/1812.00910, doi:10.1109/SP.2019.00065.
[265] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas

Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE Symposium on Security & Privacy*, IEEE S&P '21, 2021. https://arxiv.org/abs/2101.04535. doi:10.48550/arXiv.2101.04535.

[266] National Cyber Security Centre. Machine learning principles. Technical report, National Cyber Security Centre, United Kingdom, 2024. Accessed: July 18, 2024. URL: https://www.ncsc.gov.uk/collection/machine-learning-principles.

[267] National Security Commission on Artificial Intelligence. Final report. https://www.nscai.gov/2021-final-report/, 2021. doi:10.48550/ARXIV.2006.03463.

[268] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 931–962. PMLR, 16–19 Mar 2021. URL: https://proceedings.mlr.press/v132/neel21a.html, doi:10.48550/arXiv.2007.02923.

[269] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I.P. Rubinstein, Udam Saini, Charles Sutton, and Kai Xia. Exploiting machine learning to subvert your spam filter. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 08)*, San Francisco, CA, April 2008. USENIX Association. URL: https://www.usenix.org/conference/leet-08/exploiting-machine-learning-subvert-your-spam-filter.

[270] J. Newsome, B. Karp, and D. Song. Polygraph: Automatically generating signatures for polymorphic worms. In *2005 IEEE Symposium on Security and Privacy (S&P)*, pages 226–241, 2005. URL: https://ieeexplore.ieee.org/document/1425070, doi:10.1109/SP.2005.15.

[271] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1415–1432, Boston, MA, August 2022. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/nguyen.

[272] Nisos. Building Trustworthy AI: Contending with Data Poisoning, retrieved December 2024. URL: https://www.nisos.com/research/building-trustworthy-ai/.

[273] NIST. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. https://airc.nist.gov/docs/NIST.AI.600-1.GenAI-Profile.ipd.pdf, 2024. NIST AI Publication (NIST AI) NIST AI 600-1 Initial Public Draft, National Institute of Standards and Technology, Gaitherburg, MD. doi:10.6028/NIST.AI.600-1.

[274] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). https://doi.org/10.6028/NIST.AI.100-1, 2023. Online.

[275] National Institute of Standards and Technology. Managing misuse risk for dual-use

foundation models. https://doi.org/10.6028/NIST.AI.800-1.ipd, 2024. Online.

[276] Parmy Olson. Faces are the next target for fraudsters, retrieved December 2024. URL: https://www.wsj.com/articles/faces-are-the-next-target-for-fraudsters-11625662828.

[277] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[278] OpenAI. Assistants api overview, 2024. Accessed: 2024-08-18. URL: https://platform.openai.com/docs/assistants/overview.

[279] OpenAI. GPT-4o System Card. Online: https://cdn.openai.com/gpt-4o-system-card.pdf, August 2024.

[280] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4949–4958, 2019. URL: https://ieeexplore.ieee.org/document/8953839, doi:10.1109/CVPR.2019.00509.

[281] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL: https://openreview.net/forum?id=TG8KACxEON, doi:10.48550/arXiv.2203.02155.

[282] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. https://arxiv.org/abs/1605.07277, 2016. doi:10.48550/ARXIV.1605.07277.

[283] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS ’17, page 506–519, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3052973.3053009.

[284] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (S&P)*, pages 582–597, 2016. URL: https://ieeexplore.ieee.org/document/7546524, doi:10.1109/SP.2016.41.

[285] V. Pareto. *Manuale di Economia Politica*. Società Editrice Libraria, Milan, 1906.

[286] V. Pareto. *Manual of Political Economy*. Augustus M. Kelley Publishers, New York, 1971. URL: https://www.loc.gov/item/05022672/.

[287] Dario Pasquini, Martin Strohmeier, and Carmela Troncoso. Neural Exec: Learning (and learning from) execution triggers for prompt injection attacks, 2024. URL: https://arxiv.org/abs/2403.03792, arXiv:2403.03792, doi:10.48550/arXiv.2403.03792.

[288] Arpita Patra, Thomas Schneider, Ajith Suresh, and Hossein Yalame. ABY2.0: Improved Mixed-Protocol secure Two-Party computation. In *30th USENIX Security*

Symposium (USENIX Security 21), pages 2165–2182. USENIX Association, August 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/patra.

[289] Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. Label sanitization against label flipping poisoning attacks. In Carlos Alzate, Anna Monreale, Haytham Assem, Albert Bifet, Teodora Sandra Buda, Bora Caglayan, Brett Drury, Eva García-Martín, Ricard Gavaldà, Stefan Kramer, Niklas Lavesson, Michael Madden, Ian Molloy, Maria-Irina Nicolae, and Mathieu Sinn, editors, *Nemesis/UrbReas/SoGood/IWAISe/GDM@PKDD/ECML*, volume 11329 of *Lecture Notes in Computer Science*, pages 5–15. Springer, 2018. URL: http://dblp.uni-trier.de/db/conf/pkdd/nemesis2018.html#PaudiceML18.

[290] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. Asleep at the keyboard? assessing the security of github copilot's code contributions, 2021. URL: https://arxiv.org/abs/2108.09293, arXiv:2108.09293, doi:10.48550/arXiv.2108.09293.

[291] R. Perdisci, D. Dagon, Wenke Lee, P. Fogla, and M. Sharif. Misleading worm signature generators using deliberate noise injection. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, Berkeley/Oakland, CA, 2006. IEEE. URL: http://ieeexplore.ieee.org/document/1623998/, doi:10.1109/SP.2006.26.

[292] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022. URL: https://arxiv.org/abs/2202.03286, doi:10.48550/arXiv.2202.03286.

[293] Neehar Peri, Neal Gupta, W. Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P. Dickerson. Deep k-nn defense against clean-label data poisoning attacks. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 55–70, Cham, 2020. Springer International Publishing. URL: https://arxiv.org/abs/1909.13374, doi:10.48550/arXiv.1909.13374.

[294] Neil Perry, Megha Srivastava, Deepak Kumar, and Dan Boneh. Do users write more insecure code with ai assistants? In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23. ACM, November 2023. URL: http://dx.doi.org/10.1145/3576915.3623157, doi:10.1145/3576915.3623157.

[295] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. Intriguing properties of adversarial ML attacks in the problem space. In *2020 IEEE Symposium on Security and Privacy (S&P)*, pages 1308–1325. IEEE Computer Society, 2020. URL: https://doi.ieeecomputersociety.org/10.1109/SP40000.2020.00073, doi:10.1109/SP40000.2020.00073.

[296] Julien Piet, Maha Alrashed, Chawin Sitawarin, Sizhe Chen, Zeming Wei, Elizabeth Sun, Basel Alomair, and David Wagner. Jatmo: Prompt injection defense by task-specific finetuning, 2024. URL: https://arxiv.org/abs/2312.17673, arXiv:2312.17673, doi:10.48550/arXiv.2312.17673.

[297] Krishna Pillutla, Galen Andrew, Peter Kairouz, H. Brendan McMahan, Alina Oprea,

and Sewoong Oh. Unleashing the power of randomization in auditing differentially private ml. In *Advances in Neural Information Processing Systems*, 2023. URL: https://arxiv.org/abs/2305.18447, doi:10.48550/arXiv.2305.18447.

[298] PromptArmor and Kai Greshake. Data exfiltration from writer.com with indirect prompt injection, 2023. URL: https://promptarmor.substack.com/p/data-exfiltration-from-writercom.

[299] Jonathan Protzenko, Bryan Parno, Aymeric Fromherz, Chris Hawblitzel, Marina Polubelova, Karthikeyan Bhargavan, Benjamin Beurdouche, Joonwon Choi, Antoine Delignat-Lavaud, Cédric Fournet, Natalia Kulatova, Tahina Ramananandro, Aseem Rastogi, Nikhil Swamy, Christoph Wintersteiger, and Santiago Zanella-Beguelin. EverCrypt: A fast, verified, cross-platform cryptographic provider. In *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, May 2020. URL: https://eprint.iacr.org/2019/757, doi:10.1109/SP40000.2020.00114.

[300] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023. URL: https://arxiv.org/abs/2310.03693, arXiv:2310.03693.

[301] Gauthama Raman M. R., Chuadhry Mujeeb Ahmed, and Aditya Mathur. Machine learning for intrusion detection in industrial control systems: Challenges and lessons from experimental evaluation. *Cybersecurity*, 4(27), 2021. URL: https://arxiv.org/abs/2202.11917, doi:10.48550/arXiv.2202.11917.

[302] Aida Rahmattalabi, Shahin Jabbari, Himabindu Lakkaraju, Phebe Vayanos, Max Izenberg, Ryan Brown, Eric Rice, and Milind Tambe. Fair influence maximization: A welfare optimization approach. In *Proceedings of the AAAI Conference on Artificial Intelligence 35th*, 2021. URL: https://arxiv.org/abs/2006.07906, doi:10.48550/arXiv.2006.07906.

[303] Adnan Siraj Rakin, Md Hafizul Islam Chowdhuryy, Fan Yao, and Deliang Fan. DeepSteal: Advanced model extractions leveraging efficient weight stealing in memories. In *2022 IEEE Symposium on Security and Privacy (S&P)*, pages 1157–1174, 2022. URL: https://ieeexplore.ieee.org/document/9833743, doi:10.1109/SP46214.2022.9833743.

[304] Dhanesh Ramachandram and Graham W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017. URL: https://ieeexplore.ieee.org/document/8103116, doi:10.1109/MSP.2017.2738401.

[305] Javier Rando and Florian Tramèr. Universal jailbreak backdoors from poisoned human feedback, 2024. URL: https://arxiv.org/abs/2311.14455, arXiv:2311.14455, doi:10.48550/arXiv.2311.14455.

[306] Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. NeMo Guardrails: A toolkit for controllable and safe LLM applications with programmable rails, 2023. URL: https://arxiv.org/abs/2310.10501, arXiv:2310.10501, doi:10.48550/arXiv.2310.10501.

[307] Johann Rehberger. Data exfiltration via markdown injection - exploiting chatgpt's webpilot plugin, May 16 2023. Embrace The Red, Accessed: 2024-08-18. URL: https://embracethered.com/blog/posts/2023/chatgpt-webpilot-data-exfil-via-markdown-injection/.

[308] SNYK Report. AI Code Security and Trust: Organizations must change their approach. https://go.snyk.io/2023-ai-code-security-report-dwn-typ.html?aliId=eyJpIjoiUDFvdzRSdHI0dm5rVktvSSIsInQiOiJxOElRU2dQdkdqQm03ZjNLSDFFVkxBPT0ifQ%253D%253D, 2023. Human Centered AI, Stanford University.

[309] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Comput. Surv.*, 56(4), November 2023. doi:10.1145/3624010.

[310] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Comput. Surv.*, 56(4), November 2023. doi:10.1145/3624010.

[311] Rishi Bommasani, et al. On the opportunities and risks of foundation models, 2024. URL: https://arxiv.org/abs/2108.07258, arXiv:2108.07258.

[312] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *ArXiv*, abs/2310.03684, 2023. URL: https://api.semanticscholar.org/CorpusID:263671542, doi:10.48550/arXiv.2310.03684.

[313] Robust Intelligence. AI Firewall, 2023. URL: https://www.robustintelligence.com/platform/ai-firewall.

[314] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020. URL: https://arxiv.org/abs/1902.02918, doi:10.48550/arXiv.1902.02918.

[315] Benjamin IP Rubinstein, Blaine Nelson, Ling Huang, Anthony D Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J Doug Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pages 1–14, 2009. doi:10.1145/1644893.1644895.

[316] Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: The crescendo multi-turn LLM jailbreak attack. *ArXiv*, abs/2404.01833, 2024. URL: https://api.semanticscholar.org/CorpusID:268856920, doi:10.48550/arXiv.2404.01833.

[317] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 5558–5567. PMLR, 2019. URL: https://arxiv.org/abs/1908.11229, doi:10.48550/arXiv.1908.11229.

[318] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1041–1056, Washington, D.C., August 2015. USENIX Association. URL: https://www.usenix.org/conference/

usenixsecurity15/technical-sessions/presentation/sabottke.

[319] Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected?, 2024. URL: https://arxiv.org/abs/2303.11156, arXiv:2303.11156, doi:10.48550/arXiv.2303.11156.

[320] Vinu Sankar Sadasivan, Shoumik Saha, Gaurang Sriramanan, Priyatham Kattakinda, Atoosa Chegini, and Soheil Feizi. Fast adversarial attacks on language models in one gpu minute, 2024. URL: https://arxiv.org/abs/2402.15570, arXiv:2402.15570, doi:10.48550/arXiv.2402.15570.

[321] Ahmed Salem, Giovanni Cherubin, David Evans, Boris Köpf, Andrew Paverd, Anshuman Suri, Shruti Tople, and Santiago Zanella-Béguelin. SoK: Let the privacy games begin! A unified treatment of data inference privacy in machine learning. https://arxiv.org/abs/2212.10986, 2022. doi:10.48550/ARXIV.2212.10986.

[322] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. https://arxiv.org/abs/2003.03675, 2020. doi:10.48550/ARXIV.2003.03675.

[323] Roman Samoilenko. New prompt injection attack on ChatGPT web version. markdown images can steal your chat data, 2023. URL: https://systemweakness.com/new-prompt-injection-attack-on-chatgpt-web-version-ef717492c5c2.

[324] Scale AI. Adversarial robustness leaderboard. https://scale.com/leaderboard/adversarial_robustness, 2024. Accessed: 2024-08-22.

[325] Oscar Schwartz. In 2016, Microsoft's racist chatbot revealed the dangers of online conversation: The bot learned language from people on Twitter—but it also learned values. https://spectrum.ieee.org/in-2016-microsofts-racist-chatbot-revealed-the-dangers-of-online-conversation, 2019. IEEE Spectrum.

[326] R. Schwartz, A. Vassilev, K. Greene, L. Perine, A. Burt, and P. Hall. Towards a Standard for Identifying and Managing Bias in Artificial Intelligence. https://doi.org/10.6028/NIST.SP.1270, 2022. Special Publication (NIST SP) 800-1270, National Institute of Standards and Technology, Gaitherburg, MD. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1270.pdf, doi:10.6028/NIST.SP.1270.

[327] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks. https://arxiv.org/abs/2006.12557, 2020. arXiv. doi:10.48550/ARXIV.2006.12557.

[328] Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Gunnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source LLMs through the embedding space, 2024. URL: https://arxiv.org/abs/2402.09063, arXiv:2402.09063, doi:10.48550/arXiv.2402.09063.

[329] Giorgio Severi, Jim Meyer, Scott Coull, and Alina Oprea. Explanation-guided backdoor poisoning attacks against malware classifiers. In *30th USENIX Security Symposium (USENIX Security 2021)*, 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/severi.

[330] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor

Dumitras, and Tom Goldstein. Poison frogs! Targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018. URL: https://arxiv.org/abs/1804.00792, doi:10.48550/arXiv.1804.00792.

[331] Shawn Shan, Arjun Nitin Bhagoji, Haitao Zheng, and Ben Y. Zhao. Poison forensics: Traceback of data poisoning attacks in neural networks. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3575–3592, Boston, MA, August 2022. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/shan.

[332] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security*, October 2016. URL: https://www.ece.cmu.edu/~lbauer/papers/2016/ccs2016-face-recognition.pdf, doi:10.1145/2976749.2978392.

[333] Vasu Sharma, Ankita Kalra, Vaibhav, Simral Chaudhary, Labhesh Patel, and LP Morency. Attend and attack: Attention guided adversarial attacks on visual question answering models. https://nips2018vigil.github.io/static/papers/accepted/33.pdf, 2018.

[334] Ryan Sheatsley, Blaine Hoak, Eric Pauley, Yohan Beugin, Michael J. Weisman, and Patrick McDaniel. On the robustness of domain constraints. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 495–515, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3460120.3484570.

[335] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021. URL: https://www.ndss-symposium.org/wp-content/uploads/ndss2021_6C-3_24498_paper.pdf.

[336] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 1354–1371. IEEE, 2022. doi:10.1109/SP46214.2022.9833647.

[337] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*, 2023. URL: https://arxiv.org/abs/2308.03825, doi:10.48550/arXiv.2308.03825.

[338] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *CoRR*, abs/2308.03825, 2023. URL: https://doi.org/10.48550/arXiv.2308.03825, arXiv:2308.03825, doi:10.48550/ARXIV.2308.03825.

[339] Xinyue Shen, Yiting Qu, Michael Backes, and Yang Zhang. Prompt stealing attacks against text-to-image generation models. *arXiv preprint arXiv:2302.09923*, 2023.

URL: https://arxiv.org/abs/2302.09923, doi:10.48550/arXiv.2302.09923.

[340] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, and Stephen Casper. Targeted latent adversarial training improves robustness to persistent harmful behaviors in LLMs, 2024. URL: https://arxiv.org/abs/2407.15549, arXiv:2407.15549, doi:10.48550/arXiv.2407.15549.

[341] Cong Shi, Tianfang Zhang, Zhuohang Li, Huy Phan, Tianming Zhao, Yan Wang, Jian Liu, Bo Yuan, and Yingying Chen. Audio-domain position-independent backdoor attack via unnoticeable triggers. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, MobiCom '22, page 583–595, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3495243.3560531.

[342] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017. URL: https://arxiv.org/abs/1610.05820, doi:10.48550/arXiv.1610.05820.

[343] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (S&P), Oakland*, 2017. URL: https://arxiv.org/abs/1610.05820, doi:10.48550/arXiv.1610.05820.

[344] Satya Narayan Shukla, Anit Kumar Sahu, Devin Willmott, and Zico Kolter. Simple and efficient hard label black-box adversarial attacks in low query budget regimes. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1461–1469, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3447548.3467386.

[345] Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. Sponge examples: Energy-latency attacks on neural networks. https://arxiv.org/abs/2006.03463, 2020. doi:10.48550/ARXIV.2006.03463.

[346] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3, January 2019. doi:10.1145/3290354.

[347] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. URL: https://arxiv.org/abs/2001.07685, doi:10.48550/arXiv.2001.07685.

[348] Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, Chandana Satya Prakash, Mukund Sridhar, Fabian Triefenbach, Apurv Verma, Gokhan Tur, and Prem Natarajan. AlexaTM 20B: Few-shot learning using a large-scale multilingual seq2seq model. https://www.amazon.science/publications/al

exatm-20b-few-shot-learning-using-a-large-scale-multilingual-seq2seq-model, 2022. Amazon.

[349] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, Baltimore, MD, August 2018. USENIX Association. URL: https://www.usenix.org/conference/woot18/presentation/eykholt, doi:10.48550/arXiv.1807.07769.

[350] Shuang Song and David Marn. Introducing a new privacy testing library in TensorFlow, 2020. URL: https://blog.tensorflow.org/2020/06/introducing-new-privacy-testing-library.html.

[351] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongreject for empty jailbreaks, 2024. URL: https://arxiv.org/abs/2402.10260, arXiv:2402.10260, doi:10.48550/arXiv.2402.10260.

[352] N. Srndic and P. Laskov. Practical evasion of a learning-based classifier: A case study. In *Proc. IEEE Security and Privacy Symposium*, 2014. URL: https://personal.utdallas.edu/~muratk/courses/dmsec_files/srndic-laskov-sp2014.pdf.

[353] U.S. AI Safety Institute Technical Staff. Strengthening ai agent hijacking evaluations, 2024. URL: https://www.nist.gov/news-events/news/2025/01/technical-blog-strengthening-ai-agent-hijacking-evaluations.

[354] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/9d7311ba459f9e45ed746755a32dcd11-Paper.pdf, doi:10.48550/arXiv.1706.03691.

[355] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. In *Advances in Neural Information Processing Systems*, 2023. URL: https://arxiv.org/abs/2305.08846, doi:10.48550/arXiv.2305.08846.

[356] Ellen Su, Anu Vellore, Amy Chang, Raffaele Mura, Blaine Nelson, Paul Kassianik, and Amin Karbasi. Extracting memorized training data via decomposition. *arXiv preprint arXiv:2409.12367*, 2024. URL: https://arxiv.org/abs/2409.12367, doi:10.48550/arXiv.2409.12367.

[357] Octavian Suciu, Scott E Coull, and Jeffrey Johns. Exploring adversarial examples in malware detection. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 8–14. IEEE, 2019. URL: https://arxiv.org/abs/1810.08280, doi:10.48550/arXiv.1810.08280.

[358] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. When does machine learning FAIL? generalized transferability for evasion and poisoning attacks. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1299–1316, 2018. URL: https://arxiv.org/abs/1803.06975, doi:10.48550/arXiv.1803.06975.

[359] Jingwei Sun, Ang Li, Louis DiValentin, Amin Hassanzadeh, Yiran Chen, and Hai Li. FL-WBC: Enhancing robustness against model poisoning attacks in federated learning from a client perspective. In *NeurIPS*, 2021. URL: https://arxiv.org/abs/2110.13864, doi:10.48550/arXiv.2110.13864.

[360] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv:1911.07963*, 2019. URL: https://arxiv.org/abs/1911.07963, doi:10.48550/arXiv.1911.07963.

[361] Anshuman Suri and David Evans. Formalizing and estimating distribution inference risks. *Proceedings on Privacy Enhancing Technologies*, 2022. URL: https://arxiv.org/abs/2109.06024, doi:10.48550/arXiv.2109.06024.

[362] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL: http://arxiv.org/abs/1312.6199, doi:10.48550/arXiv.1312.6199.

[363] Rahim Taheri, Reza Javidan, Mohammad Shojafar, Zahra Pooranian, Ali Miri, and Mauro Conti. On defending against label flipping attacks on malware detection systems. *CoRR*, abs/1908.04473, 2019. URL: http://arxiv.org/abs/1908.04473, arXiv:1908.04473, doi:10.48550/arXiv.1908.04473.

[364] Azure AI Red Team. Pyrit: The python risk identification tool for generative ai. https://github.com/Azure/PyRIT, 2024. Accessed: 2024-08-18.

[365] The Llama Team. The LLaMA3 Herd of Models, 2024. URL: https://arxiv.org/abs/2407.21783, doi:10.48550/arXiv.2407.21783.

[366] The White House. Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence. https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/, October 2023. The White House.

[367] T. Ben Thompson and Michael Sklar. Breaking circuit breakers. URL: https://confirmlabs.org/posts/circuit_breaking.html.

[368] T. Ben Thompson and Michael Sklar. Fluent student-teacher redteaming, 2024. URL: https://arxiv.org/abs/2407.17447, arXiv:2407.17447, doi:10.48550/arXiv.2407.17447.

[369] Anvith Thudi, Ilia Shumailov, Franziska Boenisch, and Nicolas Papernot. Bounding membership inference. https://arxiv.org/abs/2202.12232, 2022. doi:10.48550/ARXIV.2202.12232.

[370] Lionel Nganyewou Tidjon and Foutse Khomh. Threat assessment in machine learning based systems. *arXiv preprint arXiv:2207.00091*, 2022. URL: https://arxiv.org/abs/2207.00091, doi:10.48550/arXiv.2207.00091.

[371] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models, 2023. URL: https://arxiv.org/abs/2302

.13971, arXiv:2302.13971, doi:10.48550/arXiv.2302.13971.

[372] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL: https://arxiv.org/abs/2307.09288, arXiv:2307.09288, doi:10.48550/arXiv.2307.09288.

[373] Florian Tramer. Detecting adversarial examples is (Nearly) as hard as classifying them. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 21692–21702. PMLR, 17–23 Jul 2022. URL: https://proceedings.mlr.press/v162/tramer22a.html.

[374] Florian Tramer, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Joern-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9561–9571. PMLR, 13–18 Jul 2020. URL: https://proceedings.mlr.press/v119/tramer20a.html, doi:10.48550/arXiv.2002.04599.

[375] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Mądry. On adaptive attacks to adversarial example defenses. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. URL: https://arxiv.org/abs/2002.08347, doi:10.48550/arXiv.2002.08347.

[376] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *USENIX Security*, 2016. URL: https://arxiv.org/abs/1609.02943, doi:10.48550/arXiv.1609.02943.

[377] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. https://arxiv.org/abs/1704.03453, 2017. doi:10.48550/ARXIV.1704.03453.

[378] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31.

Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/280cf18baf4311c92aa5a042336587d3-Paper.pdf, doi:10.48550/arXiv.1811.00636.

[379] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL: https://openreview.net/forum?id=SyxAb30cY7, doi:10.48550/arXiv.1805.12152.

[380] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. In *ICLR*, 2019. URL: https://openreview.net/forum?id=HJg6e2CcK7.

[381] U.K. AI Safety Institute. Advanced ai evaluations: May update, 2024. Accessed: 2024-08-18. URL: https://www.aisi.gov.uk/work/advanced-ai-evaluations-may-update.

[382] Kush R. Varshney. *Trustworthy Machine Learning*. Independently Published, Chappaqua, NY, USA, 2022. URL: https://www.trustworthymachinelearning.com/.

[383] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. URL: https://arxiv.org/abs/1706.03762, doi:10.48550/arXiv.1706.03762.

[384] Sridhar Venkatesan, Harshvardhan Sikka, Rauf Izmailov, Ritu Chadha, Alina Oprea, and Michael J. De Lucia. Poisoning attacks and data sanitization mitigations for machine learning models in network intrusion detection systems. In *MILCOM*, pages 874–879. IEEE, 2021. URL: https://ieeexplore.ieee.org/document/9652916, doi:10.1109/MILCOM52596.2021.9652916.

[385] Sameer Wagh, Shruti Tople, Fabrice Benhamouda, Eyal Kushilevitz, Prateek Mittal, and Tal Rabin. FALCON: honest-majority maliciously secure framework for private deep learning. In *Proceedings on Privacy Enhancing Technologies (PoPETs) 2021, Issue 1*, 20201.

[386] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. *arXiv preprint arXiv:1908.07125*, 2019. URL: https://arxiv.org/abs/1908.07125, doi:10.48550/arXiv.1908.07125.

[387] Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training LLMs to prioritize privileged instructions, 2024. URL: https://arxiv.org/abs/2404.13208, arXiv:2404.13208, doi:10.48550/arXiv.2404.13208.

[388] Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on NLP models. In *NAACL*, 2021. URL: https://arxiv.org/abs/2010.12563, doi:10.48550/arXiv.2010.12563.

[389] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning, 2023. URL: https://arxiv.org/abs/2305.00944, arXiv:2305.00944, doi:10.48550/arXiv.2305.00944.

[390] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng,

and Ben Y. Zhao. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, San Francisco, CA, USA, May 2019. IEEE. URL: https://ieeexplore.ieee.org/document/8835365/, doi:10.1109/SP.2019.00031.

[391] Haotao Wang, Tianlong Chen, Shupeng Gui, Ting-Kuei Hu, Ji Liu, and Zhangyang Wang. Once-for-All Adversarial Training: In-Situ Tradeoff between Robustness and Accuracy for Free. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada*, 2020. URL: https://arxiv.org/abs/2010.11828, doi:10.48550/arXiv.2010.11828.

[392] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. In *NeurIPS*, 2020. URL: https://arxiv.org/abs/2007.05084, doi:10.48550/arXiv.2007.05084.

[393] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024. URL: http://dx.doi.org/10.1007/s11704-024-40231-1, doi:10.1007/s11704-024-40231-1.

[394] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Formal security analysis of neural networks using symbolic intervals. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1599–1614, Baltimore, MD, August 2018. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/wang-shiqi.

[395] Wenxiao Wang, Alexander Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 22769–22783. PMLR, 2022. URL: https://proceedings.mlr.press/v162/wang22m.html, doi:10.48550/arXiv.2202.02628.

[396] Wenxiao Wang, Alexander J Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (Deterministic) finite aggregation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22769–22783. PMLR, 17–23 Jul 2022. URL: https://proceedings.mlr.press/v162/wang22m.html, doi:10.48550/arXiv.2202.02628.

[397] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1924–1933. Computer Vision Foundation / IEEE, 2021. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Enhancing_the_Transferability_of_Adversarial_Attacks_Through_Va

riance_Tuning_CVPR_2021_paper.html, doi:10.1109/CVPR46437.2021.00196.
[398] Yanting Wang, Wei Zou, and Jinyuan Jia. FCert: Certifiably robust few-shot classification in the era of foundation models. In *Proc. IEEE Security and Privacy Symposium*, 2024. URL: https://arxiv.org/abs/2404.08631, doi:10.48550/arXiv.2404.08631.
[399] Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: A dataset for evaluating safeguards in LLMs, 2023. URL: https://arxiv.org/abs/2308.13387, arXiv:2308.13387, doi:10.48550/arXiv.2308.13387.
[400] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? *arXiv preprint arXiv:2307.02483*, 2023.
[401] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. doi:10.1609/aaai.v33i01.33018973.
[402] Zhipeng Wei, Jingjing Chen, Xingxing Wei, Linxi Jiang, Tat-Seng Chua, Fengfeng Zhou, and Yu-Gang Jiang. Heuristic black-box adversarial attacks on video recognition models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12338–12345, 2020. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6918, doi:10.48550/arXiv.1911.09449.
[403] Lilian Weng. Adversarial attacks on latent language models, 2023. URL: https://lilianweng.github.io/posts/2023-10-25-adv-attack-llm/.
[404] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y. Zhao. Backdoor attacks against deep learning systems in the physical world. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6202–6211, 2020. URL: https://arxiv.org/abs/2006.14580, doi:10.48550/arXiv.2006.14580.
[405] Simon Willison. The dual LLM pattern for building AI assistants that can resist prompt injection, 2023. Accessed: 2024-08-22. URL: https://simonwillison.net/2023/Apr/25/dual-llm-pattern/.
[406] Yotam Wolf, Noam Wies, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models. *ArXiv*, abs/2304.11082, 2023. URL: https://api.semanticscholar.org/CorpusID:258291526, doi:10.48550/arXiv.2304.11082.
[407] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16913–16925. Curran Associates, Inc., 2021. URL: https://proceedings.neurips.cc/paper/2021/file/8cbe9ce23f42628c98f80fa0fac8b19a-Paper.pdf, doi:10.48550/arXiv.2110.14430.
[408] Fangzhou Wu, Ning Zhang, Somesh Jha, Patrick McDaniel, and Chaowei Xiao. A new

era in LLM security: Exploring security concerns in real-world LLM-based systems, 2024. URL: https://arxiv.org/abs/2402.18649, arXiv:2402.18649.

[409] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F. Naughton. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 355–370, 2016. doi:10.1109/CSF.2016.32.

[410] Yuhao Wu, Franziska Roesner, Tadayoshi Kohno, Ning Zhang, and Umar Iqbal. SecGPT: An execution isolation architecture for LLM-based systems, 2024. URL: https://arxiv.org/abs/2403.04960, arXiv:2403.04960, doi:10.48550/arXiv.2402.18649.

[411] Zhen Xiang, David J. Miller, and George Kesidis. Post-training detection of backdoor attacks for two-class and multi-attack scenarios. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL: https://openreview.net/forum?id=MSgB8D4Hy51, doi:10.48550/arXiv.2201.08474.

[412] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning*, pages 1689–1698, 2015. URL: https://arxiv.org/abs/1804.07933, doi:10.48550/arXiv.1804.07933.

[413] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper/2020/file/44feb0096faa8326192570788b38c1d1-Paper.pdf, doi:10.48550/arXiv.1904.12848.

[414] Weilin Xu, Yanjun Qi, and David Evans. Automatically evading classifiers. In *Proceedings of the 2016 Network and Distributed Systems Symposium*, pages 21–24, 2016. URL: https://www.cs.virginia.edu/~evans/pubs/ndss2016/.

[415] Xiaojun Xu, Xinyun Chen, Chang Liu, Anna Rohrbach, Trevor Darrell, and Dawn Song. Fooling vision and language models despite localization and attention mechanism. https://arxiv.org/abs/1709.08693, 2017. doi:10.48550/ARXIV.1709.08693.

[416] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. Detecting AI trojans using meta neural analysis. In *IEEE Symposium on Security and Privacy, S&P 2021*, pages 103–120, United States, May 2021. URL: https://ieeexplore.ieee.org/document/9519467, doi:10.1109/SP40001.2021.00034.

[417] Karren Yang, Wan-Yi Lin, Manash Barman, Filipe Condessa, and Zico Kolter. Defending multimodal fusion models against single-source adversaries. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Xplore, 2022. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9578130, doi:10.48550/ARXIV.2206.12714.

[418] Limin Yang, Zhi Chen, Jacopo Cortellazzi, Feargus Pendlebury, Kevin Tu, Fabio Pierazzi, Lorenzo Cavallaro, and Gang Wang. Jigsaw puzzle: Selective backdoor attack to subvert malware classifiers. *CoRR*, abs/2202.05470, 2022. URL: https://arxiv.or

g/abs/2202.05470, arXiv:2202.05470, doi:10.48550/arXiv.2202.05470.

[419] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL: https://arxiv.org/abs/2210.03629, arXiv:2210.03629, doi:10.48550/arXiv.2210.03629.

[420] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 2041–2055, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3319535.3354209.

[421] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 3093–3106, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3548606.3560675.

[422] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium*, CSF '18, pages 268–282, 2018. https://arxiv.org/abs/1709.01604. URL: https://arxiv.org/abs/1709.01604, doi:10.48550/arXiv.1709.01604.

[423] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *ICML*, 2018. URL: https://arxiv.org/abs/1803.01498, doi:10.48550/arXiv.1803.01498.

[424] Youngjoon Yu, Hong Joo Lee, Byeong Cheon Kim, Jung Uk Kim, and Yong Man Ro. Investigating vulnerability to adversarial examples on multimodal data fusion in deep learning. https://arxiv.org/abs/2005.10987, 2020. Online. doi:10.48550/ARXIV.2005.10987.

[425] Andrew Yuan, Alina Oprea, and Cheng Tan. Dropout attacks. In *IEEE Symposium on Security and Privacy (S&P)*, 2024. URL: https://arxiv.org/abs/2309.01614, doi:10.48550/arXiv.2309.01614.

[426] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, page 363–375, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3372297.3417880.

[427] Santiago Zanella-Beguelin, Lukas Wutschitz, Shruti Tople, Ahmed Salem, Victor Rühle, Andrew Paverd, Mohammad Naseri, Boris Köpf, and Daniel Jones. Bayesian estimation of differential privacy. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 40624–40636. PMLR, 23–29 Jul

2023. URL: https://proceedings.mlr.press/v202/zanella-beguelin23a.html, doi:10.48550/arXiv.2206.05199.

[428] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models, 2021. URL: https://arxiv.org/abs/2106.02636, arXiv:2106.02636, doi:10.48550/arXiv.2106.02636.

[429] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022. URL: https://openreview.net/forum?id=MeeQkFYVbzW, doi:10.48550/arXiv.2110.03735.

[430] Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents, 2024. URL: https://arxiv.org/abs/2403.02691, arXiv:2403.02691, doi:10.48550/arXiv.2403.02691.

[431] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, feb 2021. doi:10.1145/3446776.

[432] Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. *ArXiv*, abs/2311.04378, 2023. URL: https://api.semanticscholar.org/CorpusID:265050535, doi:10.48550/arXiv.2311.04378.

[433] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 09–15 Jun 2019. URL: https://proceedings.mlr.press/v97/zhang19p.html, doi:10.48550/arXiv.1901.08573.

[434] Ruisi Zhang, Seira Hidano, and Farinaz Koushanfar. Text revealer: Private text reconstruction via model inversion attacks against transformers. *arXiv preprint arXiv:2209.10505*, 2022. URL: https://arxiv.org/abs/2209.10505, doi:10.48550/arXiv.2209.10505.

[435] Su-Fang Zhang, Jun-Hai Zhai, Bo-Jun Xie, Yan Zhan, and Xin Wang. Multimodal representation learning: Advances, trends and challenges. In *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1–6. IEEE, 2019. URL: https://api.semanticscholar.org/CorpusID:209901378, doi:10.1109/ICMLC48188.2019.8949228.

[436] Susan Zhang, Mona Diab, and Luke Zettlemoyer. Democratizing access to large-scale language models with OPT-175B. https://ai.facebook.com/blog/democratizing-access-to-large-scale-language-models-with-opt-175b/, 2022. Meta AI.

[437] Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. Leakage of dataset properties in Multi-Party machine learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2687–2704. USENIX Association, August 2021. URL: https://www.us

enix.org/conference/usenixsecurity21/presentation/zhang-wanrong, doi:10.48550/arXiv.2006.07267.

[438] Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3), apr 2020. doi:10.1145/3374217.

[439] Yiming Zhang and Daphne Ippolito. Prompts should not be seen as secrets: Systematically measuring prompt extraction attack success. *arXiv preprint arXiv:2307.06865*, 2023. URL: https://arxiv.org/abs/2307.06865, doi:10.48550/arXiv.2307.06865.

[440] Yuhao Zhang, Aws Albarghouthi, and Loris D'Antoni. Bagflip: A certified defense against data poisoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL: https://openreview.net/forum?id=ZidkM5b92G, doi:10.48550/arXiv.2205.13634.

[441] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26429–26446. PMLR, 17–23 Jul 2022. URL: https://proceedings.mlr.press/v162/zhang22w.html, doi:10.48550/arXiv.2206.10341.

[442] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference attacks against graph neural networks. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/zhang-zhikun.

[443] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against GANs. In *Proceedings of Network and Distributed System Security*, NDSS, 2022. URL: https://arxiv.org/abs/2111.07608, doi:10.48550/arXiv.2111.07608.

[444] Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7614–7623. PMLR, 09–15 Jun 2019. URL: https://proceedings.mlr.press/v97/zhu19a.html, doi:10.48550/arXiv.1905.05897.

[445] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL: https://arxiv.org/abs/1909.08593, arXiv:1909.08593, doi:10.48550/arXiv.1909.08593.

[446] Giulio Zizzo, Chris Hankin, Sergio Maffeis, and Kevin Jones. Adversarial machine learning beyond the image domain. In *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, New York, NY, USA, 2019. Association for Com-

puting Machinery. doi:10.1145/3316781.3323470.

[447] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers, 2024. URL: https://arxiv.org/abs/2406.04313, arXiv:2406.04313, doi:h10.48550/arXiv.2406.04313.

[448] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. URL: https://arxiv.org/abs/2307.15043, doi:10.48550/arXiv.2307.15043.

[449] Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. Poisonedrag: Knowledge poisoning attacks to retrieval-augmented generation of large language models, 2024. URL: https://arxiv.org/abs/2005.11401, arXiv:2402.07867, doi:10.48550/arXiv.2005.11401.

Appendix A. Glossary

Clicking on the page number at the end of a definition will navigate to the page where the term is used.

A

adversarial example A modified testing sample that induces misclassification or misbehavior of a machine learning model at deployment time. ix, 6

adversarial machine learning Attacks that exploit the statistical, data-based nature of machine learning systems. xii, 1

agent Software programs that can interact with their environment, receive information, and undertake self-directed actions in service of a larger, externally-specified goal. 1, 35, 37, 39, 50, 52, 54

area under the curve A measure of the ability of a classifier to distinguish between classes in machine learning. A higher AUC means that a model performs better when distinguishing between the two classes. AUC measures the entire two-dimensional area under the RECEIVER OPERATING CHARACTERISTIC (ROC) curve. 30

attribute inference attacks An attack against machine learning models that infers sensitive attributes of a training data record, given partial knowledge about the record. 7

availability breakdown In the AML context, a disruption of the ability of other users or processes to obtain timely and reliable access to an AI system's outputs or functionality. 6, 39

B

backdoor pattern A transformation or insertion applied to a data sample that triggers an adversary-specified behaviour in a model that has been subject to a backdoor poisoning attack. For example, in computer vision, an adversary could poison a model such that the insertion of a square of white pixels induces a desired target label. 6, 22, 107

backdoor poisoning attack A poisoning attack that causes a model to perform an adversary-selected behaviour in response to inputs that follow a particular BACKDOOR PATTERN. 6, 42

C

classification The task of predicting which of a set of discrete categories an input belongs to. 5

convolutional neural networks A class of feed-forward neural networks that include at least one convolutional layer, referred to as CNNs. In convolutional layers, feature detectors (known as kernels or filters) detect specific features across the input data. CNNs are primarily used for processing grid-like data, such as images, and are particularly effective for tasks like image classification, object detection, and image segmentation. 5, 31

D

data confidentiality A well-established concept in cybersecurity referring to the protection of sensitive information from unauthorized access and disclosure. 7

data poisoning A POISONING ATTACKS in which an adversary controls part of the training data. 5, 36, 37, 40, 111

data privacy attacks Attacks against machine learning models that extract sensitive information about training data. 7

data reconstruction Privacy attacks that reconstruct sensitive data in a model's training data from aggregate information. 7, 28

deployment stage The stage of the machine learning pipeline in which a model is deployed into a live or real-world environment for use, such as being integrated into an enterprise application or made available to end users through an API. 5, 37, 38

diffusion models A class of latent variable generative models consisting of three major components: a forward process, a reverse process, and a sampling procedure. The goal of the diffusion model is to learn a diffusion process that generates the probability distribution of a given dataset. It is widely used in computer vision on a variety of tasks, including image denoising, inpainting, super-resolution, and image generation. 34

direct prompt injection A DIRECT PROMPTING ATTACK in which the attacker exploits PROMPT INJECTION. 43, 110

direct prompting attack In the generative AI context, an attack conducted by the primary user of the system through QUERY ACCESS (e.g., as opposed to through RESOURCE CONTROL). 34, 43, 108, 110

discriminative A type of machine learning method that learns to discriminate between classes. 5

E

energy-latency attack An attack that exploits the performance dependency on hardware and model optimizations to negate the effects of hardware optimizations, increase computational latency, increase hardware temperature, and massively increase the amount of energy consumed. 6, 8

ensemble learning A type of a meta machine learning approach that combines the predictions of several models to improve performance. 5

expectation over transformation A method for strengthening adversarial examples to remain adversarial under image transformations that occur in the real world, such as angle and viewpoint changes. EOT models these perturbations within the optimization procedure. Rather than optimizing the log-likelihood of a single example, EOT uses a chosen distribution of transformation functions that take an input controlled by the adversary to the "true" input perceived by the classifier. 16

F

federated learning A type of machine learning in which a model is trained in a decentralized fashion using multiple data sources without pooling or combining the data in any centralized location. Federated learning allows entities or devices to collaboratively train a global model by exchanging model updates without directly sharing the data that each entity controls. 5, 31

feedforward neural networks Artificial neural networks in which the connections between nodes is from one layer to the next and do not form a cycle. 31

fine-tuning The process of adapting a pre-trained model to perform specific tasks or specialize in a particular domain. This phase follows the initial pre-training phase and involves further training the model on task-specific data. This is often a supervised learning task. 37

fine-tuning circumvention Fine-tuning to remove model refusal behaviour or other model-level safety interventions. 41

formal methods A mathematically rigorous technique for the specification, development, and verification of software systems. 18

foundation model In generative AI, models trained on broad data using SELF-SUPERVISED LEARNING that can be adapted such as through fine-tuning for a variety of downstream tasks [311]. 111

functional attack An adversarial attack that is optimized for a set of data in a domain rather than per data point. 13, 23

G

generative adversarial networks A machine learning framework in which two neural networks contest with each other in the form of a zero-sum game, where one agent's gain is another agent's loss. A GAN learns to generate new data with the same statistics as the training set. See [143] for further details. 31, 34

generative pre-trained transformer (GPT) A family of machine learning models based on the transformer architecture [383] that are pre-trained through SELF-SUPERVISED LEARNING on large data sets of unlabelled text. This is the current predominant architecture for large language models. 34

graph neural network A neural network designed to process graph-structured data. GNNs perform optimizable transformations on graph attributes (e.g., nodes, edges, global context) while preserving graph symmetries such as permutation invariance. GNNs utilize a "graph-in, graph-out" architecture that takes an input graph with information and progressively transforms it into an output graph with the same connectivity as that of the input graph. 31

H

hidden Markov model A Markov model in which the system being modeled is assumed to be a Markov process with unobservable states. The model provides an observable process whose outcomes are influenced by the outcomes of a Markov model in a known way. An HMM can be used to describe the evolution of observable events that depend on internal factors that are not directly observable. In machine learning, it is assumed that the internal state of a model is hidden but not its hyperparameters. 31

I

indirect prompt injection A type of PROMPT INJECTION executed through RESOURCE CONTROL rather than through user-provided input as in a DIRECT PROMPT INJECTION. 39–41, 50

integrity violation In the AML context, an AI system being forced to misperform against its intended objectives, producing outputs or predictions that align with the attacker's objective. 6, 40

J

jailbreak A DIRECT PROMPTING ATTACK intended to circumvent restrictions placed on model outputs, such as circumventing refusal behaviour to enable misuse. 34, 38, 42, 43, 52

L

label flipping A type of data poisoning attack in which an adversary is restricted to changing the training labels. 20

label limit A capability with which an attacker does not control the labels of training samples in supervised learning. 8

logistic regression A type of linear classifier that predicts the probability of an observation being part of a class. 5

M

machine unlearning A technique that involves selectively removing the influences of specific training data points from a trained machine learning model, such as to remove unwanted capabilities or knowledge in a foundation model, or to enable a user to request the removal of their records from a model. Efficient approximate unlearning techniques may not require retraining the ML model from scratch. 33

membership-inference attack A data privacy attack to determine whether a data sample was part of the training set of a machine learning model. 7, 28

misuse enablement In the AML context, a circumvention of technical restrictions imposed by the AI system's owner on its use, such as restrictions designed to prevent a GenAI system from producing outputs that could cause harm to others. 40

model control A capability with which an attacker can control the machine learning model parameters. 8, 37, 41, 111

model extraction A type of privacy attack that extracts details of the model architecture and/or parameters. 7, 28, 31, 40, 41, 47

model poisoning A POISONING ATTACKS which operates through MODEL CONTROL. 5, 6, 37, 41, 111

model privacy attacks An attack against machine learning models to extract sensitive information about the model. 7

multimodal models A model that processes and relates information from multiple sensory modalities that each represent primary human channels of communication and sensation, such as vision and touch. 58

O

out-of-distribution Data that was collected at a different time and possibly under different conditions or in a different environment than the data collected to train the model. 56

P

poisoning attacks Adversarial attacks in which an adversary interferes with a model during its TRAINING STAGE, such as by inserting malicious training data (DATA POISONING) or modifying the training process itself (MODEL POISONING). 5, 108, 111

pre-training A component of the TRAINING STAGE in which a model learns general patterns, features, and relationships from vast amounts of unlabeled data, such as through SELF-SUPERVISED LEARNING. Pre-training can equip models with knowledge of general features or patterns which may be useful in downstream tasks (see FOUNDATION MODEL), and can be followed with additional training or fine-tuning that specializes the model for a specific downstream task. 37

privacy compromise In the AML context, the unauthorized access of restricted or proprietary information that is part of an AI system, including information about a model's training data, weights or architecture; or sensitive information that the model accesses such as the knowledge base of a GenAI RETRIEVAL-AUGMENTED GENERATION (RAG) application. 7, 40

prompt extraction An attack that tries to divulge the system prompt or other information in the context of a large language model that would normally be hidden from a user. 38, 41

prompt injection An attack which exploits the concatenation of untrusted input with a prompt constructed by a higher-trust party such as the application designer. 38, 41, 108, 110

property inference A data privacy attack that infers a global property about the training data of a machine learning model. 7

Q

query access A capability with which an attacker can issue queries to a trained machine learning model and obtain predictions or generations. 8, 40, 108

R

receiver operating characteristic (ROC) A curve that plots the true positive rate versus the false positive rate for a classifier. 107

red teaming in the AI context, means a structured testing effort, often adopting adversarial methods, to find flaws and vulnerabilities in an AI system, including unforeseen or undesirable system behaviors or potential risks associated with the misuse of the system. [366]. 60

regression A type of supervised machine learning model that is trained on data, including numerical labels (i.e., response variables). Types of regression algorithms include linear regression, polynomial regression, and various non-linear regression methods. 5

reinforcement learning A type of machine learning in which a model learns to optimize its behavior according to a reward function by interacting with and receiving feedback from an environment. 5

resource control A capability in which an attacker controls one or more external resources consumed by a machine learning model at inference time, particularly for GenAI systems such as retrieval-augmented generation applications. 41, 50, 108, 110

retrieval-augmented generation (RAG) A type of GenAI system in which a model is paired with a separate information retrieval system (or ”knowledge base”). Based on a user query, the RAG system identifies relevant information within the knowledge base and provides it to the GenAI model in context for the model to use in formulating its response. RAG systems allow the internal knowledge of a GenAI model to be modified without the need for retraining. 1, 35, 37, 38, 40, 46, 50, 111

rowhammer attack A software-based fault-injection attack that exploits dynamic random-access memory disturbance errors via user-space applications and allows the attacker to infer information about certain victim secrets stored in memory cells. Mounting this attack requires the attacker to control a user-space unprivileged process that runs on the same machine as the victim’s machine learning model. 31

S

self-supervised learning A type of machine learning that relies on generating implicit labels from unstructured data rather than relying on explicit, human-created labels. Self-supervised learning tasks are constructed to allow the true labels to be automatically inferred from the training data (enabling the use of large-scale training data) and to require models to capture essential features or relationships within the data to solve them. For example, a common self-supervised learning task is providing a model with partial data with the task to accurately generate the remainder. 109, 111

semi-supervised learning A type of machine learning in which a small number of training samples are labeled, while the majority are unlabeled. 5

shadow model A model that imitates the behavior of the target model. The training datasets and the truth about membership in these datasets are known for these models.

Typically, the attack model is trained on the labeled inputs and outputs of the shadow model. 25

side channel Allows an attacker to infer information about a secret by observing the non-functional characteristics of a program (e.g., execution time or memory) or measuring or exploiting the indirect coincidental effects of the system or its hardware (e.g., power consumption variation, electromagnetic emanations) while the program is executing. Most commonly, such attacks aim to exfiltrate sensitive information, including cryptographic keys. 31

source code control A capability with which an attacker controls the source code of a machine learning algorithm. 8

supervised learning A type of machine learning in which a model learns to predict explicit (often human-generated) labels or output values for data. 5

support vector machines Models that implement a decision function in the form of a hyperplane that serves to separate (i.e., classify) observations that belong to one class from another based on patterns of information about those observations (i.e., features). 5, 6, 31

system prompt Application-specific instructions provided in-context to a GenAI system by the model developer or application designer. System prompts are typically prepended to other input, and may be higher-trust than other forms of input. 38, 43, 46, 52

T

targeted poisoning attack A poisoning attack that changes the prediction on a small number of targeted samples. 6, 42

testing data control A capability with which an attacker controls the testing data input to the machine learning model. 8

training data control A capability in which an attacker controls some or all of the training data of a machine learning model. 7, 40

training data extraction The ability of an attacker to extract the training data of a generative model by prompting the model with specific inputs. 7, 46

training stage The stage of a machine learning pipeline in which a model learns parameters that minimize its error against an objective function based on training data. 5, 37, 111

trojan In the machine learning context, a malicious modification to a model that is difficult to detect, may appear harmless, but that can alter the intended function of the system upon a signal from an attacker to cause a malicious behavior desired by the attacker. For Trojan attacks to be effective, the trigger must be rare in the normal operating environment so that it does not affect the normal effectiveness of the AI and raise the suspicions of users. In the machine learning context, trojan may be used interchangeably with backdoor pattern. 2

U

unsupervised learning A type of machine learning in which a model learns based on patterns in unlabeled data, such as learning a function to cluster or group data points. 5