NIST AI 100-2e2023 ipd

Adversarial Machine Learning

A Taxonomy and Terminology of Attacks and Mitigations

Alina Oprea
Apostol Vassilev

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AI.100-2e2023.ipd

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

NIST AI 100-2e2023 ipd

Adversarial Machine Learning

A Taxonomy and Terminology of Attacks and Mitigations

Alina Oprea
Northeastern University

Apostol Vassilev
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AI.100-2e2023.ipd

March 2023

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

NIST Technical Series Policies
Copyright, Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Supersedes Draft NIST IR 8269 (October 2019) DOI https://doi.org/10.6028/NIST.IR.8269-draft

How to cite this NIST Technical Series Publication:
Oprea A, Vassilev A, (2023) Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. (National Institute of Standards and Technology, , Gaithersburg, MD) NIST Artificial Intelligence (AI) NIST AI 100-2e2023 ipd.
https://doi.org/10.6028/NIST.AI.100-2e2023.ipd

NIST Author ORCID iDs
Alina Oprea: 0000-0002-4979-5292
Apostol Vassilev: 0000-0002-9081-3042

Public Comment Period
March 08, 2023 - September 30, 2023

Submit Comments
ai-100-2@nist.gov

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

Abstract

This NIST NIST AI report develops a taxonomy of concepts and defines terminology in the field of adversarial machine learning (AML). The taxonomy is built on survey of the AML literature and is arranged in a conceptual hierarchy that includes key types of ML methods and lifecycle stage of attack, attacker goals and objectives, and attacker capabilities and knowledge of the learning process. The report also provides corresponding methods for mitigating and managing the consequences of attacks and points out relevant open challenges to take into account in the lifecycle of AI systems. The terminology used in the report is consistent with the literature on AML and is complemented by a glossary that defines key terms associated with the security of AI systems and is intended 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 and understanding of the rapidly developing AML landscape.

Keywords

artificial intelligence; machine learning; attack taxonomy; evasion; data poisoning; privacy breach; attack mitigation; data modality; trojan attack, backdoor attack; chatbot.

NIST AI Reports (NIST 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 . . . iv
Background . . . iv
Trademark Information . . . iv
How to read this document . . . v
Executive Summary . . . 1
1. Introduction . . . 3
2. Attack Classification . . . 6
  2.1. Stages of Learning . . . 7
  2.2. Attacker Goals and Objectives . . . 8
  2.3. Attacker Capabilities . . . 9
  2.4. Attacker Knowledge . . . 10
  2.5. Data Modality . . . 11
3. Evasion Attacks and Mitigations . . . 13
  3.1. White-Box Evasion Attacks . . . 14
  3.2. Black-Box Evasion Attacks . . . 16
  3.3. Transferability of Attacks . . . 17
  3.4. Mitigations . . . 17
4. Poisoning Attacks and Mitigations . . . 20
  4.1. Availability Poisoning . . . 20
  4.2. Targeted Poisoning . . . 22
  4.3. Backdoor Poisoning . . . 23
  4.4. Model Poisoning . . . 26
5. Privacy Attacks . . . 28
  5.1. Data Reconstruction . . . 28
  5.2. Memorization . . . 29
  5.3. Membership Inference . . . 29
  5.4. Model Extraction . . . 30
  5.5. Property Inference . . . 31
  5.6. Mitigations . . . 32
6. Discussion and Remaining Challenges . . . 33

6.1. Trade-Offs Between the Attributes of Trustworthy AI 35
6.2. Multimodal Models: Are They More Robust? 35
6.3. Beyond Models and Data 36
A. Appendix: Glossary 61

List of Figures

Fig. 1. **Taxonomy of attacks on AI systems.** 6

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 a result of an extensive literature review, conversations with experts from the area of 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 NIST 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]The term 'practice guide,' 'guide,' 'guidance' or the like, in the context of this paper, is a consensus-created, informative reference intended for voluntary use; it 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 have the force or effect of law.

How to read this document

This document uses terms such as 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:

- The document includes a series of blue callout boxes that highlight interesting nuances and important takeaways.
- Terms that are used but not defined/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 examples) 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.

Acknowledgments

The authors wish to thank colleagues from the U.S. Department of Homeland Security (DHS), National Security Agency (NSA), Federal Bureau of Investigations (FBI), Office of the Director of National Inteligence (ODNI)), the Federal Office for Information Security, Germany (BSI), academia (MIT, Georgia Tech), and industry (Google, Software Engineering Institute) who responded to our call and submitted comments to the draft version of this paper. The received comments and suggested references were essential to improving the paper and the future direction of this work. We also want to thank the many people who assisted in updating the document, including our NIST colleagues who took the time to provide their constructive feedback.

Author Contributions

Authors contributed equally and are listed in alphabetical order.

Executive Summary

This NIST AI report is intended to be a step toward developing a taxonomy and terminology of adversarial machine learning (AML), which in turn may aid in securing applications of artificial intelligence (AI) against adversarial manipulations of AI systems. The components of an AI system include – at a minimum – the data, model, and processes for training, testing, and deploying the machine learning (ML) models and the infrastructure required for using them. The data-driven approach of ML introduces additional security and privacy challenges in different phases of ML operations besides the classical security and privacy threats faced by most operational systems. These security and privacy challenges include the potential for adversarial manipulation of training data, adversarial exploitation of model vulnerabilities to adversely affect the performance of ML classification and regression, and even malicious manipulations, modifications or mere interaction with models to exfiltrate sensitive information about people represented in the data or about the model itself. Such attacks have been demonstrated under real-world conditions, and their sophistication and potential impact have been increasing steadily. AML is concerned with studying the capabilities of attackers and their goals, as well as the design of attack methods that exploit the vulnerabilities of ML during the development, training, and deployment phase of the ML life cycle. AML is also concerned with the design of ML algorithms that can withstand these security and privacy challenges. When attacks are launched with malevolent intent, the robustness of ML refers to mitigations intended to manage the consequences of such attacks.

This report adopts the notions of security, resilience, and robustness of ML systems from the NIST AI Risk Management Framework [169]. 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 (the level of risk that is acceptable to organizations or society) because it is highly contextual and application/use-case specific. This general notion of risk offers a useful approach for assessing and managing the security, resilience, and robustness of AI system components. Quantifying these likelihoods is beyond the scope of this document. Correspondingly, the taxonomy of AML is defined with respect to the following four dimensions of AML risk assessment: (i) learning method and stage of the ML life cycle process when the attack is mounted, (ii) attacker goals and objectives, (iii) attacker capabilities, (iv) and attacker knowledge of the learning process and beyond.

The spectrum of effective attacks against ML is wide, rapidly evolving, and covers all phases of the ML life cycle – from design and implementation to training, testing, and finally, to deployment in the real world. The nature and power of these attacks are different and can exploit not just vulnerabilities of the ML models but also weaknesses of the infrastructure in which the AI systems are deployed. Although AI system components may also be adversely affected by various unintentional factors, such as design and implementation flaws and data or algorithm biases, these factors are not intentional attacks. Even though these factors might be exploited by an adversary, they are not within the scope of the literature on AML or this report.

This document defines a taxonomy of attacks and introduces terminology in the field of AML. The taxonomy is built on a survey of the AML literature and is arranged in a conceptual hierarchy that includes key types of ML methods and life cycle stages of attack, attacker goals and objectives, and attacker capabilities and knowledge of the learning process. The report also provides corresponding methods for mitigating and managing the consequences of attacks and points out relevant open challenges to take into account in the life cycle of AI systems. The terminology used in the report is consistent with the literature on AML and is complemented by a glossary that defines key terms associated with the security of AI systems 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 and understanding for the rapidly developing AML landscape. Like the taxonomy, the terminology and definitions are not intended to be exhaustive but rather to aid in understanding key concepts that have emerged in AML literature.

1. Introduction

Artificial intelligence (AI) systems [164] are on a global multi-year accelerating expansion trajectory. These systems are being developed by and widely deployed into the economies of numerous countries, leading to the emergence of AI-based services for people to use in many spheres of their lives, both real and virtual [56]. Advances in the generative capabilities of AI in text and images are directly impacting society at unprecedented levels. As these systems permeate the digital economy and become inextricably 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 [169] and in the taxonomy of AI Trustworthiness [166].

However, despite the significant progress that AI and machine learning (ML) have made in a number of different application domains, these technologies are also vulnerable to attacks that can cause spectacular failures with dire consequences. For example, in computer vision applications to image classification, well-known cases of adversarial perturbations of input images have caused autonomous vehicles to swerve into the opposite direction lane and the misclassification of stop signs as speed limit signs, the disappearance of critical objects from images, and even the misidentification of people wearing glasses in high-security settings [75, 116, 193, 206]. Similarly, in the medical field where more and more ML models are being deployed to assist doctors, there is the potential for medical record leaks from ML models that can expose deeply personal information [8, 103]. Attackers can also manipulate the training data of ML algorithms, thus making the AI system trained on it vulnerable to attacks [190]. Scraping of training data from the Internet also opens up the possibility of hackers poisoning the data to create vulnerabilities that allow for security breaches down the pipeline.

Large language models (LLMs) [27, 50, 61, 154, 205, 257] are also becoming an integral part of the Internet infrastructure. LLMs are being used to create more powerful online search, help software developers write code, and even power chatbots that help with customer service. With the exception of BLOOM [154], most of the companies developing such models do not release detailed information about the data sets that have been used to build their language models, but these data sets inevitably include some sensitive personal information, such as addresses, phone numbers, and email addresses. This creates serious risks for user privacy online. The more often a piece of information appears in a data set, the more likely a model is to leak it in response to random or specifically designed queries or prompts. This could perpetuate wrong and harmful associations with damaging consequences for the people involved and bring additional security and safety concerns [34, 147].

As ML models continue to grow in size, many organizations rely on pre-trained models that could either be used directly for prediction or be fine-tuned with new datasets to enable different predictive tasks. This creates opportunities for malicious modifications of pre-trained models by inserting TROJANS to enable attackers to compromise the model availability, force incorrect processing, or leak the data when instructed [91].

This report offers guidance for the development of:

- Standardized terminology in AML to be used by the ML and cybersecurity communities;
- A taxonomy of the most widely studied and effective attacks in AML, including evasion, poisoning, and privacy attacks; and
- A discussion of potential mitigations in AML that have withstood the test of time and limitations of some of the existing mitigations.

As AML is a fast evolving field, we envision the need to update the report regularly as new developments emerge on both the attack and mitigation fronts.

> The goal of this report is not to provide an exhaustive survey of all literature on AML. In fact, this by itself is an almost impossible task as a search on arXiv for AML articles in 2021 and 2022 yielded more than 5000 references. Rather, this report provides a categorization of attacks and their mitigations, starting with the three main types of attacks: 1) evasion, 2) data and model poisoning, and 3) data and model privacy.

Historically, modality-specific ML modeling technology has emerged for each input modality (e.g., text, images, speech, tabular data), each of which is susceptible to domain-specific attacks. For example, the attack approaches for image classification tasks do not directly translate to attacks against natural language processing (NLP) models. Recently, the transformer technology from NLP has entered the computer vision domain [67]. In addition, multimodal ML has made exciting progress in many tasks, and there have been attempts to use multimodal learning as a potential mitigation of single-modality attacks [244]. However, powerful simultaneous attacks against all modalities in a multimodal model have also emerged [44, 194, 242]. The report discusses attacks against all viable learning methods (e.g., supervised, unsupervised, semi-supervised, federated learning, reinforcement learning) across multiple data modalities.

Fundamentally, the machine learning methodology used in modern AI systems is susceptible to attacks through the public APIs that the model provides and against the platforms on which they are deployed. This report focuses on the former and considers the latter to be out of scope. Attackers can breach the confidentiality and privacy protections of the data and model by simply exercising the public interfaces of the model and supplying data inputs that are within the acceptable range. In this sense, the challenges facing AML are similar to those facing cryptography. Modern cryptography relies on algorithms that are secure in an information-theoretic sense. Thus, people need to focus only on implementing them robustly and securely, which is no small task by itself. Unlike cryptography, there are no information-theoretic security proofs for the widely used machine learning algorithms.

As a result, many of the advances in developing mitigations against different classes of attacks tend to be empirical in nature.

This report is organized as follows. Section 2 introduces the taxonomy of attacks. The taxonomy is organized by first defining the broad categories of attacker objectives/goals. Based on that, we define the categories of capabilities the adversary must be able to leverage to achieve the corresponding objectives. Then, we introduce specific attack classes for each type of capability. Sections 3, 4, and 5 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 6 discusses the remaining challenges in the field.

2. Attack Classification

Figure 1 introduces a taxonomy of attacks in adversarial machine learning. The attacker's objectives are shown as disjointed circles with the attacker's goal at the center of each circle: **Availability** breakdown, **Integrity** violations, 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 requiring same capabilities for reaching the same objective are shown in a single callout. Related attack classes that require different capabilities for reaching the same objective are connected with dotted lines.

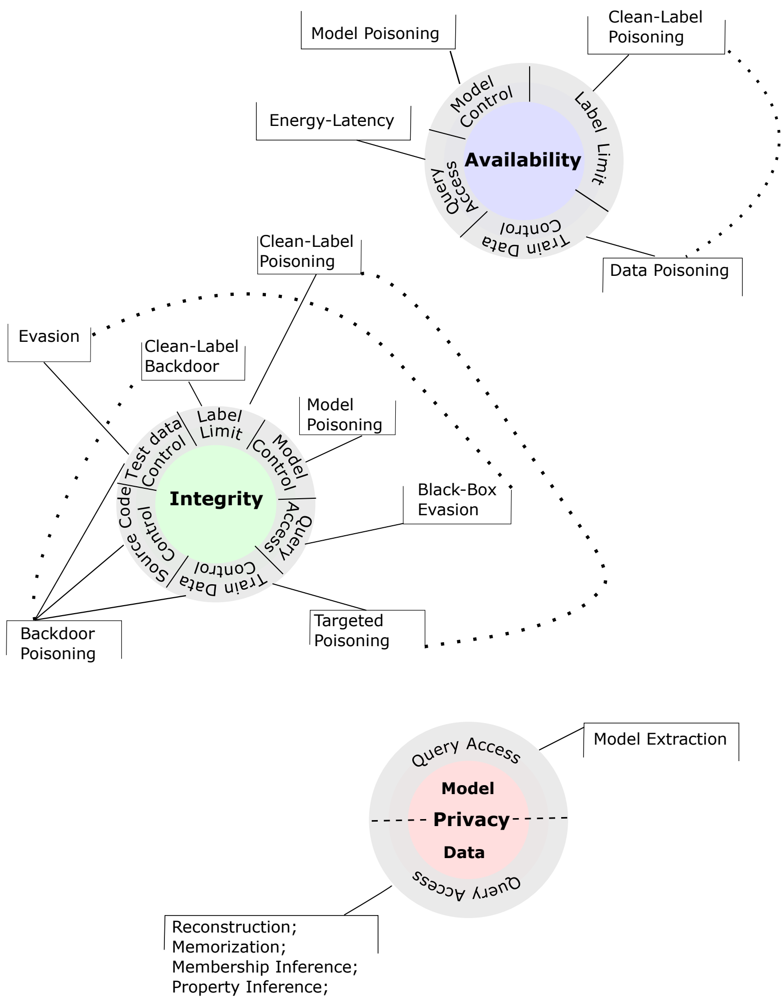

Fig. 1. Taxonomy of attacks on AI systems.

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 [23, 211], and the goal here is to create a standard terminology for adversarial attacks on ML that unifies existing work.

2.1. Stages of Learning

Machine learning involves a TRAINING STAGE, in which a model is learned, and a DEPLOYMENT STAGE, in which the model is deployed on new, unlabeled data samples to generate predictions. In the case of SUPERVISED LEARNING in the training stage labeled training data is given as input to a training algorithm 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 LOGISTIC REGRESSION, in which the predicted labels or *response variables* are continuous.

ML models may be GENERATIVE (i.e., learn the distribution of training data and generate similar examples, such as generative adversarial metworks [GAN] and large language models [LLM]) or DISCRIMINATIVE (i.e., learn only a decision boundary, such as LOGISTIC REGRESSION, SUPPORT VECTOR MACHINES, and CONVOLUTIONAL NEURAL NETWORKS).

Other learning paradigms in the ML literature are 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, which performs an aggregation of model updates; ENSEMBLE LEARNING which is an approach in machine learning that seeks better predictive performance by combining the predictions from multiple models.

Adversarial machine learning literature predominantly considers adversarial attacks against AI systems that could occur at either the training stage or the ML deployment stage. During the ML 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 ML 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. Attacks during the ML training stage are called POISONING ATTACKS [21]. In a DATA POISONING attack [21, 94], an adversary controls a subset of the training data by either inserting or modifying training samples. In a MODEL POISONING attack [137], 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, 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. Two different types of attacks can be mounted at testing/deployment time. First, evasion attacks modify testing samples to create ADVERSARIAL EXAMPLES [19, 93, 215], which are similar to the original sample (according to certain distance metrics) but alter the model predictions to the attacker's choices. Second, privacy attacks, such as membership inference [199] and data reconstruction [66], are typically mounted by attackers with query access to an ML model. They could be further divided into data privacy attacks and model privacy attacks.

2.2. 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 (i.e., availability, integrity, confidentiality): availability breakdown, integrity violations, 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. An AVAILABILITY ATTACK is an indiscriminate attack against ML in which the attacker attempts to break down the performance of the model at testing/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 attacks via query access. Data poisoning availability attacks have been proposed for SUPPORT VECTOR MACHINES [21], linear regression [111], and even neural networks [140, 160], while model poisoning attacks have been designed for neural networks [137] and federated learning [6]. Recently, ENERGY-LATENCY ATTACKS that require only black-box access to the model have been developed for neural networks across many different tasks in computer vision and NLP [202].

Integrity Violations. An INTEGRITY ATTACK targets the integrity of an ML model's output, resulting in incorrect predictions performed by an ML model. An attacker can cause an integrity violation by mounting an evasion attack at testing/deployment time or a poisoning attack at training time. Evasion attacks require the modification of testing samples to create adversarial examples that are mis-classified by the model to a different class, while remaining stealthy and imperceptible to humans [19, 93, 215]. Integrity attacks via poisoning can be classified as TARGETED POISONING ATTACKS [89, 192], BACKDOOR POISONING ATTACKS [94], and MODEL POISONING [6, 17, 77]. 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 [137] and federated learning [6, 17].

Privacy Compromise. Attackers might be interested in learning information about the training data (resulting in DATA PRIVACY attacks) or about 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 [66] (inferring content or features of training data), MEMBERSHIP-INFERENCE ATTACKS [99, 200] (inferring the presence of data in the training set), data MEMORIZATION [33, 34] (ability to extract training data from generative models), and PROPERTY INFERENCE [85] (inferring properties about the training data distribution). MODEL EXTRACTION is a model privacy attack in which attackers aim to extract information about the model [108].

2.3. Attacker Capabilities

An adversary might leverage six types of capabilities to achieve their objectives, as shown in the outer layer of the objective circles in Figure 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 may utilize this to 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 the 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 – a realistic poisoning scenario, 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: When the ML model is managed by a cloud provider (using Machine Learning as a Service – MLaaS), the attacker might submit queries to the model

and receive predictions (either labels or model confidences). This capability is used by black-box evasion attacks, energy-latency attacks, and all privacy attacks.

Note that even if an attacker does not have the ability to modify training/testing data, source code, or model parameters, access to these are still crucial for mounting white-box attacks. See Section 2.4 for more details on attacker knowledge.

Figure 1 connects each attack class with the capabilities required to mount the attack. For instance, backdoor attacks that cause integrity violations require control of training data 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.

2.4. Attacker Knowledge

Another dimension for attack classification is how much knowledge the attacker has about the ML system. There are three main types of attacks: white-box, black-box, and gray-box.

White-box attacks. These assume that the attacker operates with *full* knowledge about the ML system, including the training data, model architecture, and model hyper-parameters. 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. Note that this definition is more general and encompasses the notion of adaptive attacks where the knowledge of the mitigations applied to the model or the system is explicitly tracked.

Black-box attacks. These attacks assume minimal knowledge about the ML system. An adversary might get 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 utilize 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. [211] 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 in applications where feature extraction is used before training an ML model, such as cybersecurity, finance, and healthcare.

2.5. Data Modality

Adversarial attacks against ML have been discovered in a range of data modalities used in many application domains. Until recently, most attacks and defenses have operated under a single modality, but a new ML trend is to use multimodal data. The taxonomy of attacks defined in Figure 1 is independent of the modality of the data in specific applications.

The most common data modalities in the adversarial ML literature include:

1. **Image:** Adversarial examples of image data modality [93, 215] 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 [94], and many privacy attacks are run on image datasets (e.g., [199]).

2. **Text:** Natural language processing (NLP) is a popular modality, and all classes of attacks have been proposed for NLP applications, including evasion [96], poisoning [48, 131], and privacy [252]. Audio systems and text generated from audio signals have also been attacked [37].

3. **Cybersecurity[2]:** The first poisoning attacks were discovered in cybersecurity for worm signature generation (2006) [176] and spam email classification (2008) [165]. Since then, poisoning attacks have been shown for malware classification, malicious PDF detection, and Android malicious app classification [191]. Evasion attacks against the same data modalities have been proposed as well: malware classification [62, 210], PDF malware classification [208, 241], and Android malicious app detection [178]. Clements et al. [57] developed a mechanism for effective generation of evasion attacks on small, weak routers in network intrusion detection. Poisoning unsupervised learning models has been shown for clustering used in malware classification [22] and network traffic anomaly detection [185].

   Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) systems are part of modern Critical Infrastructure (CI) such as power grids, power plants (nuclear, fossil fuel, renewable energy), water treatment plants, oil refineries, etc. ICS are an attractive target for adversaries because of the potential for highly consequential disruptions of CI [38, 127]. The existence of targeted stealth attacks has led to the development of defense-in-depth mechanisms for their detection and mitigation. Anomaly detection based on data-centric approaches allows automated feature learning through ML algorithms. However, the application of ML to such problems comes with specific challenges related to the need for a very low false negative and low false positive rates, ability to catch zero-day attacks, account for plant operational drift, etc. This challenge is compounded by the fact that trying to accommodate all these together makes ML models susceptible to adversarial attacks [122, 179, 264].

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

4. **Tabular data:** Numerous attacks against ML models working on tabular data in finance, business, and healthcare applications have been demonstrated. For example, poisoning availability attacks have been shown against healthcare and business applications [111]; privacy attacks have been shown against healthcare data [248]; and evasion attacks have been shown against financial applications [90].

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 [244], but other papers show that multimodal models themselves could be vulnerable to attacks mounted on all modalities at the same time [44, 194, 242]. See Section 6.2 for additional discussion.

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

3. Evasion Attacks and Mitigations

The discovery of evasion attacks against machine learning models has generated increased interest in adversarial machine learning, leading to significant growth in this research space over the last decade. In an evasion attack, the adversary's goal is to generate adversarial examples, which are defined as testing samples whose classification can be changed at deployment time to an arbitrary class of the attacker's choice with only minimal perturbation [215]. Early known instances of evasion attacks date back to 1988 with the work of Kearns and Li [119], and to 2004, when Dalvi et al. [60], and Lowd and Meek [139] demonstrated the existence of adversarial examples for linear classifiers used in spam filters. Adversarial examples became even more intriguing to the research community when Szedegy et al. [215] showed that deep neural networks used for image classification can be easily manipulated, and adversarial examples were visualized. In the context of image classification, the perturbation of the original sample must be small so that a human cannot observe the transformation of the input. Therefore, 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.

In 2013, Szedegy et al. [215] and Biggio et al. [19] 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 [5, 36, 143]. 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 [47], discrete optimization [155], and Bayesian optimization [201], as well as *transferability*, which involves the white-box generation of adversarial examples on a different model architecture before transferring them to the target model [172, 173, 222]. Cybersecurity and image classifications were the first application domains that showcased evasion attacks. However, with the increasing interest in adversarial machine learning, ML technology used in many other application domains went under scrutiny, including speech recognition [37], natural language processing [115], and video classification [133, 235].

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, a process that appears to iterate in perpetuity. Mitigations need to be evaluated against strong adaptive attacks, and guidelines for the rigorous evaluation of newly proposed mitigation techniques have been established [59, 220]. The most promising directions for mitigating the critical threat of evasion attacks are adversarial training [93, 143] (iteratively generating and inserting adversarial examples with their correct labels at training time); certified techniques, such as randomized smoothing [58] (evaluating ML prediction under noise); and formal verification techniques [88, 118] (applying formal method techniques to verify the model's output). Nevertheless, these methods come with 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 [219, 224, 255]. Similarly, there are trade-offs between a model's robustness and fairness guarantees [41].

This section discusses white-box and black-box evasion attack techniques, attack transferability, and the potential mitigation of adversarial examples in more detail.

3.1. White-Box Evasion Attacks

There are several optimization-based methods for designing evasion attacks that generate adversarial examples at small distances from the original testing samples. There are also several choices for distance metrics, universal evasion attacks, and physically realizable attacks, as well as examples of evasion attacks developed for multiple data modalities, including NLP, audio, video, and cybersecurity domains.

Optimization-based methods. Szedegy et al. [215] and Biggio et al. [19] 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, in which the adversarial example's class is selected by the attacker, or untargeted, in which the adversarial examples are misclassified to any other incorrect class.

Szedegy et al. [215] 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. [19] considered the setting of a binary classifier with malicious and benign classes with 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. [19] apply their method to linear classifiers, kernel SVM, and multi-layer perceptrons, Szedegy et al. [215] show the existence of adversarial examples on deep learning models used for image classification. Goodfellow et al. [93] 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. [124].

Subsequent work on generating adversarial examples 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:

1. DeepFool is an untargeted evasion attack for ℓ_2 norms, which uses a linear approximation of the neural network to construct the adversarial examples [157].
2. 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 [36] 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 [174].
3. The Projected Gradient Descent (PGD) attack [143] 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. [156] showed how to construct small universal perturbations (with respect to some norm), which 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. 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 machine learning systems that become feasible in the physical world. One of the first physically realizable attacks in the literature is the attack on facial recognition systems by Sharif et al. [193]. 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. [76] 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.

Other data modalities. In computer vision applications, adversarial examples must 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. The concept of adversarial examples has been extended to other domains, such as audio, video, natural language processing (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 are discussed below:

- **Audio:** Carlini and Wagner [37] 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 [235] and dense attacks that perturb all of the frames in a video [133]. The goal of the attacker is to change the classification label of the video.
- **NLP:** Jia and Liang [115] developed a methodology for generating adversarial NLP examples. This pioneering work was followed by many advances in developing adversarial attacks on NLP models (see a comprehensive survey on the topic [259]). Recently, La Malfa and Kwiatkowska [125] 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.
- **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 [53]. FENCE has been applied to two network security applications: malicious domain detection and malicious network traffic classification. Sheatsley et al. [195] propose 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 apply the technique to network intrusion detection and phishing classifiers. Both papers observe that attacks from continuous domains cannot be readily applied in constrained environments, as they result in infeasible adversarial examples. Pierazzi et al. [178] discuss the difficulty of mounting feasible evasion attacks in cyber security due to constraints in feature space and the challenge of mapping attacks from feature space to problem space. They formalize evasion attacks in problem space and construct feasible adversarial examples for Android malware.

3.2. Black-Box Evasion Attacks

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 machine learning as a service (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 [47, 105]. Other optimization techniques include discrete optimization [155], natural evolution strategies [104], and random walks [161].

- **Decision-based attacks:** In this more restrictive setting, attackers obtain only 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 [25], which was extended with an improved gradient estimation to reduce the number of queries in the HopSkipJumpAttack [46]. More recently, several optimization methods search for the direction of the nearest decision boundary (the OPT attack [51]), use sign SGD instead of binary searches (the Sign-OPT attack [52]), or use Bayesian optimization [201].

The main 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 [201].

3.3. Transferability of Attacks

Another method for generating adversarial attacks under restrictive threat models is via transferability of 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. [172, 173] 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 [135, 222, 234].

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 [93, 156, 222]. Demontis et al. [63] 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.

3.4. Mitigations

Mitigating evasion attacks is challenging because adversarial examples are widespread in a variety of ML model architectures and application domains, as discussed above. 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 [106]. 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 [35]. Recent work shows that detecting adversarial examples is as difficult as building a defense [218]. Therefore, this direction for mitigating adversarial examples is similarly challenging when designing defenses.
- The Obfuscated Gradients attack [5] was specifically designed to defeat several proposed defenses that mask the gradients using the ℓ_0 and ℓ_∞ distance metrics. It relies on a new technique, Backward Pass Differentiable Approximation, which approximates the gradient during the backward pass of backpropagation. It bypasses seven proposed defenses.
- Tramèr et al. [220] described a methodology for designing adaptive attacks against proposed defenses and circumvented 13 existing defenses. They advocate 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 proved resilient and have the potential to provide mitigation against evasion attacks:

1. **Adversarial training:** Introduced by Goodfellow et al. [93] and further developed by Madry et al. [143], adversarial training is a general method that augments the 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. Interestingly, adversarial training results in models with more semantic meaning than standard models [224], 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. [128] and further improved by Cohen et al. [58], 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 depends on the radius of the ℓ_2 ball and the characteristics of the training data and model).
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 [118]. AI^2 is the first verification method applicable to convolutional neural networks using abstract interpretation techniques [88]. These methods have been extended and scaled up to larger networks in follow-up verification systems, such as DeepPoly [203], ReluVal [232], and Fast Geometric Projections (FGP) [84]. Formal verification techniques have significant potential for certifying neural network robustness, but their main limitations are their lack of scalability, computational cost, and restriction in the type of supported operations.

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.

4. Poisoning Attacks and Mitigations

Another relevant threat against machine learning systems is the risk of adversaries mounting poisoning attacks, which are broadly defined as adversarial attacks during the training stage of the ML algorithm. Poisoning attacks have a long history in cybersecurity, as the first known poisoning attack was developed for worm signature generation in 2006 [176]. Since then, poisoning attacks have been studied extensively in several application domains: computer security (for spam detection [165]), network intrusion detection [226], vulnerability prediction [187], malware classification [191, 239]), computer vision [89, 94, 192], natural language processing [48, 131, 228], and tabular data in healthcare and financial domains [111]. Recently, poisoning attacks have gained more attention in industrial applications as well. A Microsoft report revealed that they are considered to be the most critical vulnerability of machine learning systems deployed in production [123].

Poisoning attacks are very powerful and can cause either an availability violation or an integrity violation. In particular, availability poisoning attacks cause indiscriminate degradation of the machine learning model on all samples, while targeted and backdoor poisoning attacks are stealthier and induce integrity violations on a small set of target samples. Poisoning attacks leverage a wide range of adversarial capabilities, such as 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 adversarial scenarios [21, 111, 239], gray-box settings [111], and black-box models [20]. This section discusses the threat of 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 as well as existing mitigations and their limitations are also discussed.

4.1. Availability Poisoning

The first poisoning attacks discovered in cybersecurity applications were availability attacks against worm signature generation and spam classifiers, which indiscriminately impact the entire machine learning model and, in essence, cause a denial-of-service attack on users of the AI system. Perdisci et al. [176] generated suspicious flows with fake invariants that mislead the worm signature generation algorithm in Polygraph [167]. Nelson et al. [165] designed poisoning attacks against Bayes-based spam classifiers, which generate spam emails that contain long sequences of words appearing in legitimate emails to induce the misclassification of spam emails. Both of these attacks were conducted under the white-box setting in which adversaries are aware of the ML training algorithm, feature representations, training datasets, and ML models. ML-based methods have been proposed for the detection of cybersecurity attacks targeting ICS. Such detectors are often retrained using data collected during system operation to account for plant operational drift of the monitored signals. This retraining procedure creates opportunities for an attacker to mimic the signals of corrupted sensors at training time and poison the learning process of the detector such that attacks remain undetected at deployment time [122].

A simple black-box poisoning attack strategy is LABEL FLIPPING, which generates training examples with a victim label selected by the adversary [20]. This method requires a large percentage of poisoning samples for mounting an availability attack, and it has been improved via optimization-based poisoning attacks introduced for the first time against SUPPORT VECTOR MACHINES (SVM) [21]. In this approach, the attacker solves a bilevel optimization problem to determine the optimal poisoning samples that will achieve the adversarial objective (i.e., maximize the hinge loss for SVM [21] or maximize the mean square error [MSE] for regression [111]). These optimization-based poisoning attacks have been subsequently designed against linear regression [111] and neural networks [160], and they 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 [63, 211].

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 case models 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 [81]. A different approach for clean-label poisoning is to use gradient alignment and minimally modify the training data [82].

Availability poisoning attacks have also been designed for unsupervised learning against centroid-based anomaly detection [120] and behavioral clustering for malware [22]. In federated learning, an adversary can mount a model poisoning attack to induce availability violations in the globally trained model [77, 196, 197]. More details on model poisoning attacks are provided in Section 4.4.

Mitigations.

Availability poisoning attacks are usually detectable by monitoring the standard performance metrics of ML models – such as precision, recall, accuracy, F1 scores, and area under the curve – as they cause a large degradation in the classifier metrics. Nevertheless, detecting these attacks during the testing or deployment stages of ML is less desirable, and existing mitigations aim to proactively prevent these attacks during the training stage to generate robust ML models. Among the existing mitigations, some generally promising techniques include:

- **Training data sanitization:** These methods leverage the insight that poisoned samples are typically different than regular training samples not controlled by adversaries. As such, data sanitization techniques are designed to clean the training set and remove the poisoned samples before the machine learning training is performed. Nelson et al. [165] propose 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 this early approach by reducing its computational complexity. Paudice et al. [175] introduced a method for label cleaning that was specifically designed for label flipping attacks. Steinhardt et al. [209] propose the use of outlier detection methods for identifying poisoned samples. Clustering methods have also been used for detecting poisoned samples [126, 216]. In the context of network intrusion detection, computing the variance of predictions made by an ensemble of multiple ML models has proven to be an effective data sanitization method [226]. Once sanitized, the datasets should be protected by cybersecurity mechanisms for dataset origin and integrity attestation [164].

- **Robust training:** An alternative approach to mitigating availability poisoning attacks is to modify the ML training algorithm and perform robust training instead of regular training. The defender can train an ensemble of multiple models and generate predictions via model voting [18, 130, 233]. Several papers apply techniques from robust optimization, such as using a trimmed loss function [65, 111]. Rosenfeld et al. [183] proposed the use of randomized smoothing for adding noise during training and obtaining certification against label flipping attacks.

4.2. Targeted Poisoning

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 the clean-label setting in which the attacker does not have access to the labeling function.

Several techniques for mounting clean-label targeted attacks have been proposed. Koh and Liang [121] showed how influence functions – a statistical method that determines the most influential training samples for a prediction – can be leveraged for creating poisoned samples in the fine-tuning setting in which a pre-trained model is fine-tuned on new data. Suciu et al. [211] 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. [192] to generate clean-label targeted poisoning for fine-tuning and end-to-end learning. ConvexPolytope [263] and BullseyePolytope [2] optimized the poisoning samples against ensemble models, which offers better advantages for attack transferability. MetaPoison [101] uses a meta-learning algorithm to optimize the poisoned samples, while Witches' Brew [89] 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 [112] 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 [29], such as MixMatch [15], FixMatch [204], and Unsupervised Data Augmentation (UDA) [240] 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. [112] showed an impossibility result for subpopulation poisoning attacks. To mitigate some of the risks associated with such attacks, cybersecurity mechanisms for dataset origin and integrity attestation [164] should be used judiciously. Ma et al. [141] proposed the use of differential privacy (DP) as a defense (which follows directly from the definition of differential privacy), but it is well known that differentially private ML models have lower accuracy than standard models. The trade-off between robustness and accuracy needs to be considered in each application. If the application has strong data privacy requirements, and differentially private training is used for privacy, then an additional benefit is protection against targeted poisoning attacks. However, the robustness offered by DP starts to fade once the targeted attack requires multiple poisoning samples (as in subpopulation poisoning attacks) because the group privacy bound will not provide meaningful guarantees for large poisoned sets.

4.3. Backdoor Poisoning

In 2017, Gu et al. [94] 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 – including the trigger or backdoor pattern – will be misclassified to the target class at testing time. Concurrently, Chen et al. [49] introduced backdoor attacks in which the trigger is blended into the training data. Follow-up work introduced the concept of clean-label backdoor attacks [225] in which the adversary is restricted in preserving 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 [246]. Backdoor Generating Network (BaN) [188] 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 are embedded throughout the image or change according to the input. For instance, Li et al. [132] used steganography algorithms to hide the trigger in the training data. Liu et al. [138] introduced a clean-label attack that uses natural reflection on images as a backdoor trigger. Wenger et al. [236] poisoned facial recognition systems by using physical objects as triggers, such as sunglasses and earrings.

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. [198] 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 natural language processing, 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. [48] introduced semantic-preserving backdoors at the character, word, and sentence level for sentiment analysis and neural machine translation applications. Li et al. [131] generated hidden backdoors against transformer models using generative language models in three NLP tasks: toxic comment detection, neural machine translation, and question answering.

- **Cybersecurity:** Early poisoning attacks in cybersecurity were designed against worm signature generation in 2006 [176] and spam detectors in 2008 [165], well before rising interest in adversarial machine learning. More recently, Severi et al. [191] 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 [245] 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, model inspection and sanitization, and also their limitations.

- **Training Data Sanitization:** Similar to poisoning availability attacks, training data sanitization can be applied to detecting backdoor poisoning attacks. For instance, outlier detection in the latent feature space [98, 177, 223] has been effective for convolutional neural networks used for computer vision applications. Activation Clustering [43] performs clustering of training data in representation space with the goal of isolating 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 that effective against stealthy poisoning attacks. Overall, this leads to a trade-off between attack success and 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. [229] 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 [100, 238]. A representative system in this class is Artificial Brain Simulation (ABS) by Liu et al. [136], which stimulates multiple neurons and measures the activations to reconstruct the trigger patterns.

- **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 [102], which is based on explainability methods to determine different features between clean and backdoored models that are subsequently used for outlier detection. DeepInspect [45] uses a conditional generative model to learn the probability distribution of trigger patterns and performs model patching to remove the trigger. Xu et al. [243] 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 [237], retraining [253], or fine-tuning [134] to restore the model's accuracy.

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. [191] showed that some of the data sanitization techniques (e.g., spectral signatures [223] and Activation Clustering [43]) are ineffective against clean-label backdoor poisoning on malware classifiers. Most 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 [243] is the high computational complexity of the training stage of the meta classifier, which requires training thousands of SHADOW MODELS. 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. [184] proposed a principal component analysis (PCA)-

based approach to mitigate poisoning attacks against PCA subspace anomaly detection method 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 [142] built an autoencoder-based intrusion detection system, assuming malicious poisoning attack instances were under 2%.

4.4. Model Poisoning

Model poisoning attacks attempt to directly modify the trained ML model to inject malicious functionality into the model. In centralized learning, TrojNN [137] 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 [77, 196].
- 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 [7, 16, 213].
- Backdoor model poisoning attacks introduce a trigger via malicious client updates to induce the misclassification of all samples with the trigger at testing time [7, 16, 213, 231]. 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 [260].

Model poisoning attacks are also possible in supply-chain scenarios where models or components of the model provided by suppliers are poisoned with malicious code.

Mitigations. To defend federated learning from model poisoning attacks, a variety of Byzantine-resilient aggregation rules have been designed and evaluated. Most of them attempt to identify and exclude the malicious updates when performing the aggregation at the server [3, 24, 28, 95, 148–150, 212, 250]. However, motivated adversaries can bypass these defenses by adding constraints in the attack generation optimization problem [7, 77, 196]. Gradient clipping and differential privacy have the potential to mitigate model poisoning attacks to some extent [7, 168, 213], but they usually decrease accuracy and do not provide complete mitigation.

Designing federated learning models that are fully robust against model poisoning attacks remains an open research problem in the community.

5. Privacy Attacks

Although privacy issues have long been a concern, privacy attacks against aggregate statistical information collected from user records started with the seminal work of Dinur and Nissim [66] on *reconstruction attacks*. The goal of reconstruction attacks is to reverse engineer private information about an individual user record or sensitive critical infrastructure data from access to aggregate statistical information. More recently, *memorization* attacks that reconstruct or regenerate the training data have been shown in the context of large generative language models, such as GPT-2 [34]. A less devastating privacy attack is that of *membership inference* in which an adversary can determine whether a particular record was included in the dataset used for computing statistical information or training a machine learning model. Membership inference attacks were first introduced by Homer et al. [99] 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 [30, 199, 249]. Another 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 [32, 40, 109, 221]. Property inference attacks [4, 42, 86, 144, 214, 258] aim to extract global information about a training dataset, such as the fraction of training examples with a certain sensitive attribute.

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

5.1. Data Reconstruction

Data reconstruction attacks are the most concerning privacy attacks as they have the ability to recover an individual's data from released aggregate statistical information. Dinur and Nissim [66] were the first to introduce reconstruction attacks that recover user data from linear statistics. Their original attack requires an exponential number of queries for reconstruction, but subsequent work has shown how to perform reconstruction with a polynomial number of queries [73]. A survey of privacy attacks, including reconstruction attacks, is given by Dwork et al. [71]. More recently, the U.S. Census Bureau performed a large-scale study on the risk of data reconstruction attacks on census data [87], 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. [83] introduced model inversion attacks that reconstruct class representatives from the training data of an ML model. While model inversion generates semantically similar images with those in the training set, it cannot directly reconstruct the training data of the model. Recently, Balle et al. [9] trained a reconstructor network that can recover a data sample from a neural network model, assuming a powerful adversary with information about all other training samples. Haim et al. [97] showed how the training data of a neural network can be reconstructed from access to the model parameters by leveraging theoretical insights about implicit bias in neural networks.

5.2. Memorization

Memorization attacks are a powerful class of techniques that allow an adversary to extract training data from generative ML models, such as language models. Carlini et al. [33] were the first to practically demonstrate memorization attacks in language models. By inserting synthetic canaries 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 memorization in large language models, such as GPT-2 [34], and showed that models with a larger capacity tend to memorize more [31].

An orthogonal line of work is analyzing the connection between memorization and generalization in ML models. Zhang et al. [254] discussed how neural networks can memorize randomly selected datasets. Feldman [79] showed that the memorization of training labels is necessary to achieving almost optimal generalization error in ML. Brown et al. [26] 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 [80].

5.3. Membership Inference

Membership inference attacks generally expose less private information about an individual than reconstruction or memorization attacks but are still of great concern when releasing aggregate statistical 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 extraction attacks [33, 34].

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. [99] 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 [72]. 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 used for classification [30, 54, 129, 199, 247, 248]. Similar to other attacks in adversarial machine learning, membership inference can be performed in white-box settings [129, 162, 186] 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 [30, 54, 199, 247, 248].

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 [114, 248]. In terms of techniques for mounting membership inference attacks, the loss-based attack by Yeom et al. [248] 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. [186] refined the loss-based attack by scaling the loss using a per-example threshold. Another popular technique introduced by Shokri et al. [199] is that of *shadow models*, which trains a meta-classifier on examples in and out of the training set obtained from 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 [30].

An intermediary method that is currently attaining state-of-the-art performance in terms of the AREA UNDER THE CURVE (AUC) metric is the LiRA attack by Carlini et al. [30], 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. [247] 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. 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 [54].

There are several public privacy libraries that offer implementations of membership inference attacks: the TensorFlow Privacy library [207] and the ML Privacy Meter [159].

5.4. Model Extraction

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. [221] on several online ML services for different ML models, including logistic regression, decision trees, and neural networks. However, Jagielski et al. [109] 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. [109] have shown that even the weaker task of extracting functionally equivalent models is *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 [32, 109, 221]. A second technique explored in a series of papers is to use learning methods for extraction. For instance, active learning [40] 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 [171]. A third technique is the use of side channel information for model extraction. Batina et al. [12] used electromagnetic side channels to recover simple neural network models, while Rakin et al. [181] recently showed how rowhammer attacks can be used for model extraction of more complex convolutional neural network architectures.

5.5. Property Inference

In property inference attacks, the attacker tries to learn global information about the training data distribution by interacting with an ML model. For instance, an attacker can determine the fraction of the training set with a certain sensitive attribute, such as 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. [4] and formalized as a distinguishing game between the attacker and the challenger training two models with different fractions of the sensitive data [214]. Property inference attacks were designed in white-box settings in which the attacker has access to the full ML model [4, 86, 214] and black-box settings in which the attacker issues queries to the model and learns either the predicted labels [144] or the class probabilities [42, 258]. These attacks have been demonstrated for hidden Markov models, Support Vector Machines [4], feed-forward neural networks [86, 144, 258], convolutional neural networks [214], federated learning models [146], generative adversarial networks [262], and graph neural networks [261]. Mahloujifar et al. [144] and Chaudhauri et al. [42] showed that poisoning the property of interest can help design a more effective distinguishing test for property inference. Moreover, Chaudhauri et al. [42] designed an efficient property size estimation attack that recovers the exact fraction of the population of interest.

Several papers have reported negative results on various mitigation strategies against these attacks, including differential privacy which was designed to reveal aggregate statistics about a dataset [42, 144]. It seems inherent that a high accuracy ML model will reveal some aggregate information about its training dataset. While property inference might not be easy to mitigate, an open problem is understanding whether these attacks pose real privacy risk to users who contribute their data to ML training.

5.6. Mitigations

The discovery of reconstruction attacks against aggregate statistical information motivated the rigorous definition of *differential privacy* (DP) [69, 70]. Differential privacy is 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 [153].

DP has been widely adopted due to several useful properties: group privacy (i.e., the extension of the definition to two datasets differing 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 that are performed on the dataset). DP mechanisms for statistical computations include the Gaussian mechanism [70], the Laplace mechanism [70], and the Exponential mechanism [145]. The most widely used DP algorithm for training ML models is DP-SGD [1], with recent improvements such as DP-FTRL [117] and DP matrix factorization [64].

By definition, DP provides mitigation against reconstruction attacks, the memorization of training data, and membership inference attacks. In fact, the definition of DP immediately implies an upper bound on the success of a membership inference attack. Tight bounds on the success of membership inference have been derived by Thudi et al. [217]. However, DP does not provide guarantees against model extraction or property inference attacks [42, 144]. One of the main challenges of using DP in practice is setting up the privacy parameters to achieve a trade-off between privacy and utility, which is typically measured in terms of accuracy for ML models. Analysis of privacy-preserving algorithms, such as DP-SGD, is often worst case, 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. Recently, a promising line of work is that of *privacy auditing* introduced by Jagielski et al. [113] with the goal of empirically measuring the actual privacy guarantees of an algorithm and determining privacy lower bounds by mounting privacy attacks. Auditing can be performed with membership inference attacks [114], but poisoning attacks are much more effective for empirical privacy auditing [113, 163].

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 exist in the literature. However, these techniques can be circumvented by motivated and well-resourced attackers and should be used with caution. We refer the reader to available practice guides for securing machine learning deployments [39, 169].

6. Discussion and Remaining Challenges

The literature on AML shows a trend of designing new attacks with higher power and stealthier behavior. The attacks considered above and those discussed in Section 6.2 illustrate this well. Moreover, Goldwasser et al. [91] recently introduced a new class of attacks: information-theoretically undetectable Trojans that can be planted in ML models. Such attacks can only be prevented or detected and mitigated by procedures that restrict and control who in the organization has access to the model throughout the life cycle and by thoroughly vetting third-party components coming through the supply chain. The NIST AI Risk Management Framework [169] offers more information on this.

One of the ongoing challenges facing the AML field is the ability to detect when the 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. Tramèr [218] has shown that designing techniques to detect adversarial examples is equivalent to robust classification, which is inherently hard to construct, up to computational complexity and a factor of 2 in the robustness radius.

Adversarial examples may be from the same data distribution on which the model is 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. [78] established useful theoretical bounds on detectability, particularly an impossibility result when there is an overlap between the in-distribution and OOD data.

Given the onslaught of powerful attacks, designing appropriate mitigations is a challenge that needs to be addressed before deploying AI systems in critical domains. This challenge is exacerbated by the lack of information-theoretically secure machine learning algorithms for many tasks in the field, as we discussed in Section 1. This implies that presently designing mitigations is an inherently ad hoc and fallible process. We refer the readers to available practice guides for securing machine learning deployments [39, 169], as well as existing guidelines for mitigating AML attacks [74].

The data and model sanitization techniques discussed in Section 4 reduce the impact of a range of poisoning attacks and should be widely used. However, they should 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 [164].

The robust training techniques discussed in Section 4 offer different approaches to providing 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 the data distributions, the ability to handle OOD inputs, more complex models, and multiple data modalities. Another challenge is applying these techniques to very large models like LLMs and generative models, which are quickly becoming targets of attacks [55].

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 into this direction [59, 190], more research and encouragement is needed to foster the creation of standardized benchmarks to allow gaining reliable insights into the actual performance of proposed mitigations.

Formal methods verification has a long history in other fields where high assurance is required, such as avionics and cryptography. The lessons learned there teach us that although the results from applying this methodology are excellent in terms of 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 mandated by regulations. Applying formal methods to neural networks has significant 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 machine learning models of interest at acceptable costs. More research is needed to extend this technology to all algebraic operations used in machine learning algorithms, to scale it up to the large models used today, and to accommodate rapid changes in the code of AI systems while limiting the costs of applying formal verification.

There is an imbalance between the large number of privacy attacks listed in Section 5 (i.e., memorization, membership inference, model extraction, and property inference) and available reliable mitigation techniques. In some sense, this is a normal state of affairs: a rapidly evolving technology gaining widespread adoption – even "hype" – which attracts the attention of adversaries, who try to expose and exploit its weaknesses before the technology has matured enough for society to assess and manage it effectively. To be sure, not all adversaries have malevolent intent. Some simply want to warn the public of potential breakdowns that can cause harm and erode trust in the technology. Additionally, not all attacks are as practical as they need to be to pose real threats to AI system deployments of interest. Yet the race between developers and adversaries has begun, and both sides are making great progress. This poses many difficult questions for the AI community of stakeholders, such as:

- What is the best way to mitigate the potential exploits of memorized data from Section 5.2 as models grow and ingest larger amounts of data?
- What is the best way to prevent attackers from inferring membership in the training set or other properties of the training data using the attacks listed in Sections 5.3 and 5.5?
- How can developers protect their ML models and associated intellectual property from the emerging threats of algebraic methods that utilize the public API of the ML

model to query and exploit its secret weights or the side-channel leakage attacks from Section 5.4? The known mechanisms of preventing large numbers of queries through the API are ineffective in configurations with anonymous or unauthenticated access to the model.

As answers to these questions become available, it is important for the community of stakeholders to develop specific guidelines to complement the NIST AI RMF [169] for use cases where privacy is of utmost importance.

6.1. Trade-Offs Between the Attributes of Trustworthy AI

The trustworthiness of an AI system depends on all of the attributes that characterize it [169]. For example, an AI system that is accurate but easily susceptible to adversarial exploits is unlikely to be trusted. Similarly, an AI system that produces harmfully biased or unfair outcomes is unlikely to be trusted even if it is robust. There are also trade-offs between explainability and adversarial robustness [107, 152]. In cases where fairness is important and privacy is necessary to maintain, the trade-off between privacy and fairness needs to be considered [110]. Unfortunately, it is not possible to simultaneously maximize the performance of the AI system with respect to these attributes. For instance, AI systems optimized for accuracy alone tend to underperform in terms of adversarial robustness and fairness [41, 68, 180, 224, 255]. Conversely, an AI system optimized for adversarial robustness may exhibit lower accuracy and deteriorated fairness outcomes [14, 230, 255].

> The full characterization of the trade-offs between the different attributes of trustworthy AI is still an open research problem that is gaining increasing importance with the adoption of AI technology in many areas of modern life.

In most cases, organizations will need to accept trade-offs between these properties and decide which of them to prioritize depending on the AI system, the use case, and potentially many other considerations about the economic, environmental, social, cultural, political, and global implications of the AI technology [169].

6.2. Multimodal Models: Are They More Robust?

MULTIMODAL MODELS have shown great potential for achieving high performance on many machine learning tasks [10, 13, 158, 182, 256]. It is natural to assume that because there is redundancy of information across the different modalities, the model should be more robust against adversarial perturbations of a single modality. However, emerging evidence from practice shows that this is not necessarily the case. Combining modalities and training the model on clean data alone does not seem to improve adversarial robustness. In addition, one of the most effective defenses against evasion attacks based on adversarial training, which is widely used in single modality applications, is prohibitively expensive in practical applications of multimodal learning. Additional effort is required to benefit from the redundant information in order to improve robustness against single modality attacks [244]. 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 [244, 251]. 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 [44, 194, 242].

> 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. Attackers in real life do not constrain themselves to attacks within a given security model but employ any attack that is available to them.

6.3. Beyond Models and Data

As pointed out in the Introduction, chatbots [50, 61, 151, 170] enabled by recent advances in deep learning have emerged as a powerful technology with great potential for numerous business applications, from entertainment to more critical fields. AI-enabled chatbots use NLP to process and respond to human input, but these chatbots have more complicated architectures than just a language model. For example, a critical element of a conversational chatbot is the dialog component whose task is to determine the purpose of the user input and identify relevant intents (i.e., establish the context for the conversation). Only then is the chatbot able to determine an appropriate response and return it to the user. Traditional attacks on chatbots have focused on overwhelming the chatbot with toxic input in order to alter its behaviour [189]. Recently, specific attacks using ”PROMPT INJECTIONS” have emerged as effective ways to trigger bad behaviour in the bot [227].

However, the design of AI systems that can communicate with humans is not just a technical problem but a deeply socio-technical challenge. In addition, the potential for attacks that could compromise the function of the dialog component and maliciously change the subject of the conversation for the unsuspecting user can lead to the chatbot offering misleading or even harmful advice. The potential harms in this case go beyond the traditional harms considered by AML and defined in Section 2.

> Despite progress in the ability of chatbots to perform well on certain tasks [170], this technology is still limited and should not be deployed in applications that require a high degree of trust in the information they generate.

As the development of AI-enabled chatbots continues and their deployment becomes more prevalent online, these concerns will come to the forefront and be pursued by adversaries to discover and exploit vulnerabilities and by companies developing the technology to improve their design and implementation to protect against such attacks.

Realistic risk management throughout the entire life cycle of the technology is critically important to identify risks and plan early corresponding mitigation approaches [169]. For example, incorporating human adversarial input in the process of training the system (i.e., red teaming) or employing reinforcement learning from human feedback appear to offer benefits in terms of making the chatbot more resilient against toxic input or prompt injections [61]. Barrett et al. [11] have developed detailed risk profiles for cutting-edge generative AI systems that map well to the NIST AI RMF [56] and should be used for assessing and mitigating potentially catastrophic risks to society that may arise from this technology.

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.

[2] 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, EuroS&P 2021, Vienna, Austria, September 6-10, 2021*, pages 159–178. IEEE, 2021.

[3] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine Stochastic Gradient Descent. In *NeurIPS*, 2018.

[4] 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, sep 2015.

[5] 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.

[6] 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.

[7] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *AISTATS*. PMLR, 2020.

[8] 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.

[9] Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[10] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy, 2017.

[11] Anthony M. Barrett, Dan Hendrycks, Jessica Newman, and Brandie Nonnecke. Actionable Guidance for High-Consequence AI Risk Management: Towards Standards Addressing AI Catastrophic Risks. https://arxiv.org/abs/2206.08966, 2022.

[12] 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.

[13] 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, aug 2022.

[14] Philipp Benz. Trade-off between accuracy, robustness, and fairness of deep classifiers. 2021.

[15] 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.

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

[17] 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.

[18] 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.

[19] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndić, 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.

[20] 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.

[21] 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.

[22] 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*, pages 27–36, 2014.

[23] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Com-*

puter and Communications Security, CCS '18, page 2154–2156, New York, NY, USA, 2018. Association for Computing Machinery.

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

[25] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, 2017.

[26] 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.

[27] 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.

[28] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *NDSS*, 2021.

[29] 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.

[30] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 2022 IEEE Symposium on Security and Privacy (SP) (SP)*, pages 1519–1519, Los Alamitos, CA, USA, may 2022. IEEE Computer Society.

[31] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2022.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proc. IEEE Security and Privacy Symposium*, 2017.

[37] 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.

[38] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 388:1–29, 2016.

[39] 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.

[40] 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.

[41] Hong Chang, Ta Duy Nguyen, Sasi Kumar Murakonda, Ehsan Kazemi, and R. Shokri. On adversarial bias and the robustness of fair machine learning. *ArXiv*, abs/2006.08669, 2020.

[42] 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.

[43] 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, 2018.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] Alesia Chernikova and Alina Oprea. FENCE: Feasible evasion attacks on neural networks in constrained environments. *ACM Transactions on Privacy and Security (TOPS) Journal*, 2022.

[54] 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.

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

[56] 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.

[57] Joseph Clements, Yuzhe Yang, Ankur Sharma, Hongxin Hu, and Yingjie Lao. Rallying adversarial techniques against deep learning for network security, 2019.

[58] 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.

[59] 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.

[60] 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.

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

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

[68] 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.

[69] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming,*

33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II, pages 1–12, 2006.

[70] 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.

[71] 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.

[72] 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.

[73] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *Annual International Cryptology Conference*, pages 469–480. Springer, 2008.

[74] 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.

[75] 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.

[76] 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.

[77] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *USENIX Security*, 2020.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

[87] Simson Garfinkel, John Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Communications of the ACM*, 62:46–53, 02 2019.

[88] 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 (SP)*, pages 3–18, 2018.

[89] 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.

[90] 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.

[91] 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.

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

[93] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[94] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[95] Rachid Guerraoui, Arsany Guirguis, Jérémy Plassmann, Anton Ragot, and Sébastien Rouault. Garfield: System support for byzantine machine learning (regular paper). In *DSN*. IEEE, 2021.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] Xijie Huang, Moustafa Alzantot, and Mani Srivastava. NeuronInspect: Detecting backdoors in neural networks via output explanations, 2019.

[103] W. Nicholson Price II. Risks and remedies for artificial intelligence in health care. https://www.brookings.edu/research/risks-and-remedies-for-artificial-intelligence-i

n-health-care/, 2019. Brookings Report.

[104] 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.

[105] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019.

[106] 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.

[107] 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.

[108] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. *High Accuracy and High Fidelity Extraction of Neural Networks*. USENIX Association, USA, 2020.

[109] 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.

[110] 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.

[111] 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 *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35, 2018.

[112] 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.

[113] 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.

[114] Bargav Jayaraman and David Evans. Evaluating differentially private machine learn-

ing in practice. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 1895–1912, USA, 2019. USENIX Association.

[115] 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.

[116] 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.

[117] 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.

[118] 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 Kuncak, editors, *Computer Aided Verification*, pages 97–117, Cham, 2017. Springer International Publishing.

[119] 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.

[120] Marius Kloft and Pavel Laskov. Security analysis of online centroid anomaly detection. *Journal of Machine Learning Research*, 13(118):3681–3724, 2012.

[121] 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.

[122] 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.

[123] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning – industry perspectives, 2020.

[124] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2016.

[125] 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.

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

classifier from poisoning attack. *CoRR*, abs/1606.01584, 2016.

[127] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

[128] 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.

[129] 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.

[130] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[131] 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.

[132] 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.

[133] 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.

[134] 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 (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 273–294. Springer Verlag, 2018. Funding Information: Acknowledgement. This research was partially supported by National Science Foundation CAREER Award #1553419. Publisher Copyright: © Springer Nature Switzerland AG 2018.; 21st International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2018 ; Conference date: 10-09-2018 Through 12-09-2018.

[135] 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.

[136] 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.

[137] 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.

[138] 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.

[139] 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.

[140] Yiwei Lu, Gautam Kamath, and Yaoliang Yu. Indiscriminate data poisoning attacks on neural networks, 2022.

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

[142] Pooria Madani and Natalija Vlajic. Robustness of deep autoencoder in intrusion detection under adversarial contamination. pages 1–8, 04 2018.

[143] 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.

[144] 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.

[145] 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.

[146] 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.

[147] Melissa Heikkilä. What does GPT-3 "know" about me? https://www.technologyreview.com/2022/08/31/1058800/what-does-gpt-3-know-about-me/, August 2022. MIT Technology Review.

[148] 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.

[149] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The Hidden Vulnerability of Distributed Learning in Byzantium. In *ICML*, 2018.

[150] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed momentum for byzantine-resilient stochastic gradient descent. In *ICLR*, 2021.

[151] Microsoft. Power virtual agents. https://powervirtualagents.microsoft.com/en-us/ai-chatbot/, 2022. Online.

[152] 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.

[153] Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[154] 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.

[155] 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.

[156] 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.

[157] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2015.

[158] 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.

[159] Sasi Kumar Murakonda and Reza Shokri. ML Privacy Meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning, 2020.

[160] 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.

[161] 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.

[162] 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.

[163] 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.

[164] National Security Commission on Artificial Intelligence. Final report. https://www.nscai.gov/2021-final-report/, 2021.

[165] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles A Sutton, J Doug Tygar, and Kai Xia. Exploiting machine learning to subvert your spam filter. 2008.

[166] Jessica Newman. A Taxonomy of Trustworthiness for Artificial Intelligence: Connecting Properties of Trustworthiness with Risk Management and the AI Lifecycle. Technical report, Center for Long Term Cybersecurity, University of California, Berkeley, 2023. Online: https://cltc.berkeley.edu/wp-content/uploads/2023/01/Taxonomy_of_AI_Trustworthiness.pdf.

[167] J. Newsome, B. Karp, and D. Song. Polygraph: automatically generating signatures for polymorphic worms. In *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pages 226–241, 2005.

[168] 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.

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

[170] OpenAI. ChatGPT: Optimizing Language Models for Dialogue. https://openai.com/blog/chatgpt/, 2022. Online.

[171] 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.

[172] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016.

[173] 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.

[174] 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 (SP)*, pages 582–597, 2016.

[175] 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.

[176] 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.

[177] 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.

[178] 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 (SP)*, pages 1308–1325. IEEE Computer Society, 2020.

[179] 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.

[180] 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.

[181] 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 (SP)*, pages 1157–1174, 2022.

[182] 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.

[183] 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.

[184] Benjamin I.P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shinghon Lau, Satish Rao, Nina Taft, and J. D. Tygar. Antidote: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC '09, page 1–14, New York,

NY, USA, 2009. Association for Computing Machinery.

[185] 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.

[186] 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.

[187] 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.

[188] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models, 2020.

[189] 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.

[190] 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.

[191] 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.

[192] 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.

[193] 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.

[194] Vasu Sharma, Ankita Kalra, Vaibhav, Simral Chaudhary, Labhesh Patel, and LP Morency. Attend and attack : Attention guided adversarial attacks on visual question answering models. 2018.

[195] 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.

[196] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing

model poisoning attacks and defenses for federated learning. In *NDSS*, 2021.

[197] 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.

[198] 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.

[199] 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.

[200] 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.

[201] 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.

[202] 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.

[203] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL), jan 2019.

[204] 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.

[205] 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/alexatm-20b-few-shot-learning-using-a-large-scale-multilingual-seq2seq-model, 2022. Amazon.

[206] 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.

[207] Shuang Song and David Marn. Introducing a new privacy testing library in TensorFlow, 2020.

[208] N. Srndic and P. Laskov. Practical evasion of a learning-based classifier: A case study. In *Proc. IEEE Security and Privacy Symposium*, 2014.

[209] 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.

[210] 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.

[211] 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.

[212] 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.

[213] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv:1911.07963*, 2019.

[214] Anshuman Suri and David Evans. Formalizing and estimating distribution inference risks. *Proceedings on Privacy Enhancing Technologies*, 2022.

[215] 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.

[216] 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.

[217] Anvith Thudi, Ilia Shumailov, Franziska Boenisch, and Nicolas Papernot. Bounding membership inference, 2022.

[218] 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.

[219] 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.

[220] 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.

[221] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *USENIX Security*, 2016.

[222] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples, 2017.

[223] 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.

[224] 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.

[225] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. In *ICLR*, 2019.

[226] 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.

[227] Brandon Vigliarolo. GPT-3 'prompt injection' attack causes bad bot manners. https://www.theregister.com/2022/09/19/in_brief_security/, 2022. The Register, Online.

[228] Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models, 2020.

[229] 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.

[230] 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.

[231] 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.

[232] 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.

[233] 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.

[234] 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.

[235] 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.

[236] 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.

[237] 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.

[238] 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.

[239] 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.

[240] 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.

[241] Weilin Xu, Yanjun Qi, and David Evans. Automatically evading classifiers. In *Proceedings of the 2016 Network and Distributed Systems Symposium*, pages 21–24, 2016.

[242] 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.

[243] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *Proceedings - 2021 IEEE Symposium on Security and Privacy, SP 2021*, Proceedings - IEEE Symposium on Security and Privacy, pages 103–120, United States, May 2021. Institute of Electrical and Electronics Engineers Inc. Funding Information: This material is based upon work

supported by the Department of Energy under Award Number DE-OE0000780. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Publisher Copyright: © 2021 IEEE.; 42nd IEEE Symposium on Security and Privacy, SP 2021 ; Conference date: 24-05-2021 Through 27-05-2021.

[244] 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.

[245] 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.

[246] 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.

[247] 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.

[248] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[249] 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.

[250] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *ICML*, 2018.

[251] 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.

[252] 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*, page 363–375. Association for Computing Machinery, New York, NY, USA, 2020.

[253] 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.

[254] 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.

[255] 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.

[256] 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.

[257] 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.

[258] 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.

[259] 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.

[260] 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.

[261] 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.

[262] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against GANs. In *Proceedings of Network and Distributed System Security*, NDSS, 2022.

[263] 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.

[264] 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 Computing Machinery.

Note: one may click on the page number shown at the end of the definition of each glossary entry to go to the page where the term is used.

A. Appendix: Glossary

adversarial examples Modified testing samples which induce mis-classification of a machine learning model at deployment time. v, 8

Area Under the Curve In ML the Area Under the Curve (AUC) is a measure of the ability of a classifier to distinguish between classes. The higher the AUC, the better the performance of the model at distinguishing between the two classes. AUC measures the entire two-dimensional area underneath the RECEIVER OPERATING CHARACTERISTICS (ROC) curve. 30

availability attack Adversarial attacks against machine learning which degrade the overall model performance. 8

backdoor pattern A trigger pattern inserted into a data sample to induce mis-classification of a poisoned model. For example, in computer vision it may be constructed from a set of neighboring pixels, e.g., a white square, and added to a specific target label. To mount a backdoor attack, the adversary first poisons the data by adding the trigger to a subset of the clean data and changing their corresponding labels to the target label. 9

backdoor poisoning attacks Poisoning attacks against machine learning which change the prediction on samples including a backdoor pattern. 8

classification Type of supervised learning in which data labels are discrete. 7

convolutional neural networks A Convolutional Neural Network (CNN) is a class of artificial neural networks whose architecture connects neurons from one layer to the next layer and includes at least one layer performing convolution operations. CNNs are typically applied to image analysis and classification. See [92] for further details. 7, 31

data poisoning Poisoning attacks in which a part of the training data is under the control of the adversary. 7

data privacy Attacks against machine learning models to extract sensitive information about training data. 9

data reconstruction Data privacy attacks which reconstruct sensitive information about training data records. 9

deployment stage Stage of ML pipeline in which the model is deployed on new data. 7

discriminative Type of machine learning methods which learn to discriminate between classes. 7

energy-latency attacks Attacks that exploit the performance dependency on hardware and model optimizations to negate the effects of hardware optimizations, increase computation latency, increase hardware temperature and massively increase the amount of energy consumed. 8

ensemble learning Type of a meta machine learning approach that combines the predictions of several models to improve the performance of the combination. 7

federated learning Type of collaborative machine learning, in which multiple users train jointly a machine learning model. 7

federated learning models Federated learning is a methodology to train a decentralized machine learning model (e.g., deep neural networks or a pre-trained large language model) across multiple end-devices without sharing the data residing on each device. Thus, the end-devices collaboratively train a global model by exchanging model updates with a server that aggregates the updates. Compared to traditional centralized learning where the data are pooled, federated learning has advantages in terms of data privacy and security but these may come as tradeoffs to the capabilities of the models learned through federated data. Other potential problems one needs to contend with here concern the trustworthiness of the end-devices and the impact of malicious actors on the learned model. 31

feed-forward neural networks A Feed Forward Neural Network is an artificial neural network in which the connections between nodes is from one layer to the next and do not form a cycle. See [92] for further details. 31

formal methods Formal methods are mathematically rigorous techniques for the specification, development, and verification of software systems. 18

generative Type of machine learning methods which learn the data distribution and can generate new examples from distribution. 7

generative adversarial networks A generative adversarial network (GAN) is a class of machine learning frameworks 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. GAN's learn to generate new data with the same statistics as the training set. See [92] for further details. 31

graph neural networks A Graph Neural Network (GNN) is an optimizable transformation on all attributes of the graph (nodes, edges, global-context) that preserves the graph symmetries (permutation invariances). GNNs utilize a "graph-in, graph-out" architecture that takes an input graph with information loaded into its nodes, edges and global-context, and progressively transform these embeddings into an output graph with the same connectivity as that of the input graph. 31

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

integrity attack Adversarial attacks against machine learning which change the output prediction of the machine learning model. 8

label flipping a type of data poisoning attack where the adversary is restricted to changing the training labels. 21

label limit Capability in which the attacker in some scenarios does not control the labels of training samples in supervised learning. 9

logistic regression Type of linear classifier that predicts the probability of an observation to be part of a class.. 7

membership-inference attacks Data privacy attacks to determine if a data sample was part of the training set of a machine learning model. 9

memorization The ability of a machine learning model to encode, remember, and potentially emit the training data. 9

model control Capability in which the attacker has control over machine learning model parameters. 9

model extraction Type of privacy attack to extract model architecture and parameters. 9

model poisoning Poisoning attacks in which the model parameters are under the control of the adversary. 8

model privacy Attacks against machine learning models to extract sensitive information about the model. 9

multimodal models Modality is associated with the sensory modalities which represent primary human channels of communication and sensation, such as vision or touch. Multimodal models process and relate information from multiple modalities. 35

out-of-distribution This term refers to 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. 33

poisoning attacks Adversarial attacks against machine learning at training time. 7

prompt injections Malicious plain text instructions to a generative AI system that uses textual instructions (a "prompt") to accomplish a task causing the AI system to generate text on a topic prohibited by the designers of the system. 36

property inference Data privacy attacks which infer global property about the training data of a machine learning model. 9

query access Capability in which the attacker can issue queries to a trained machine learning model and obtain predictions. 9

Receiver Operating Characteristics (ROC) In ML the Receiver Operating Characteristics (ROC) curve plots true positive rate versus false positive rate for a classifier. 61

reinforcement learning Type of machine learning in which an agent interacts with the environment and learns to take actions which optimize a reward function. 7

rowhammer attacks Rowhammer is a software-based fault-injection attack that exploits DRAM 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 attacker's control of a user-space unprivileged process that runs on the same machine as the victim's ML model. 31

semi-supervised learning Type of machine learning in which a small number of training samples are labeled, while the majority are unlabeled. 7

shadow models Shadow models imitate the behavior of the target model. The training datasets and thus the ground 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 models. 25

side channel side channels allow an attacker to infer information about a secret by observing nonfunctional characteristics of a program, such as execution time or memory or by measuring or exploiting indirect coincidental effects of the system or its hardware, like 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 Capability in which the attacker has control over the source code of the machine learning algorithm. 9

supervised learning Type of machine learning methods based on labeled data. 7

Support Vector Machines A Support Vector Machine implements a decision function in the form of a hyperplane that serves to separate (i.e., classify) observations belonging to one class from another based on patterns of information about those observations (i.e., features). . 7, 8, 21, 31

targeted poisoning attacks Poisoning attacks against machine learning which change the prediction on a small number of targeted samples. 8

testing data control Capability in which the attacker has control over the testing data input to the machine learning model. 9

training data control Capability in which the attacker has control over a part of the training data of a machine learning model. 9

training stage Stage of machine learning pipeline in which the model is trained using training data. 7

trojans A malicious code/logic inserted into the code of a software or hardware system, typically without the knowledge and consent of the organization that owns/develops the system, that is difficult to detect and may appear harmless, but can alter the intended function of the system upon a signal from an attacker to cause a malicious behavior desired by the attacker. 3

unsupervised learning Type of machine learning methods based on unlabeled data. 7