Withdrawn Draft

Warning Notice

The attached draft document has been withdrawn and is provided solely for historical purposes. It has been superseded by the document identified below.

Withdrawal Date	March 8, 2023
Original Release Date	October 30, 2019

Superseding Document

Status	Initial Public Draft (ipd)
Series/Number	NIST AI 100-2e2023 ipd
Title	Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations
Publication Date	March 8, 2023
DOI	https://doi.org/10.6028/NIST.AI.100-2e2023.ipd
CSRC URL	https://csrc.nist.gov/publications/detail/white-paper/2023/03/08/adversarial-machine-learning-taxonomy-and-terminology/draft
Additional Information	

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

Draft NISTIR 8269

A Taxonomy and Terminology of Adversarial Machine Learning

Elham Tabassi
Kevin J. Burns
Michael Hadjimichael
Andres D. Molina-Markham
Julian T. Sexton

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8269-draft

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

Draft NISTIR 8269

A Taxonomy and Terminology of Adversarial Machine Learning

Elham Tabassi
National Institute of Standards and Technology
Information Technology Laboratory

Kevin J. Burns
Michael Hadjimichael
Andres D. Molina-Markham
Julian T. Sexton
National Cybersecurity Center of Excellence
The MITRE Corporation

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8269-draft

October 2019

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report 8269
35 pages (October 2019)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8269-draft

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Public comment period: *October 30, 2019* through *December 16, 2019*

National Institute of Standards and Technology
Attn: National Cybersecurity Center of Excellence (NCCoE)
100 Bureau Drive (Mail Stop 2002) Gaithersburg, Maryland 20899-2000
Email: ai-nccoe@nist.gov

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

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL 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 for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

This NIST Interagency/Internal Report (NISTIR) is intended as a step toward securing applications of Artificial Intelligence (AI), especially against adversarial manipulations of Machine Learning (ML), by developing a taxonomy and terminology of Adversarial Machine Learning (AML). Although AI also includes various knowledge-based systems, the data-driven approach of ML introduces additional security challenges in training and testing (inference) phases of system operations. AML is concerned with the design of ML algorithms that can resist security challenges, the study of the capabilities of attackers, and the understanding of attack consequences.

This document develops a taxonomy of concepts and defines terminology in the field of AML. The taxonomy, built on and integrating previous AML survey works, is arranged in a conceptual hierarchy that includes key types of attacks, defenses, and consequences. The terminology, arranged in an alphabetical glossary, defines key terms associated with the security of ML components of an AI system. Taken together, the terminology and taxonomy are intended to inform future standards and best practices for assessing and managing the security of ML components, by establishing a common language and understanding of the rapidly developing AML landscape.

Keywords

adversarial; artificial intelligence; attack; cybersecurity; defense; evasion; information technology; machine learning; oracle; poisoning.

Acknowledgments

The authors wish to thank the many people who assisted with the development of this document, including our NIST colleague Tim McBride. We would also like to thank the technical review team from The MITRE Corporation for their support on this effort: Dan Aiello, Lashon Booker, Ron Ferguson, Chuck Howell, Keith Manville, Joseph Mikhail, Scott Musman, Colin Shea-Blymyer, Anne Townsend, and Michael Zoracki. Also, we would like to thank the technical review from our academic team: Edward Colbert and Laura Freeman from Virginia Tech, and Tim Oates from University of Maryland, Baltimore County.

Audience

The main audience for this document is researchers and practitioners in the field of machine learning (artificial intelligence). Researchers and practitioners in adversarial machine learning will find this useful for choosing the correct and standardized terminology to be used in their own reports. Machine learning researchers may also benefit by understanding the relationships between adversarial attacks, defenses, and consequences, and by referencing the definitions of standardized terminology.

Trademark Information

All trademarks and registered trademarks belong to their respective organizations.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

    i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or
    ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: ai-nccoe@nist.gov

Table of Contents

List of Figures

List of Tables

1 Introduction

This NIST Interagency/Internal Report (NISTIR) is intended as a step toward securing applications of Artificial Intelligence (AI), especially against adversarial manipulations of Machine Learning (ML), by developing a taxonomy and terminology of Adversarial Machine Learning (AML). AI refers to computer systems able to perform tasks that normally require human intelligence, such as image classification and speech recognition. ML refers to the components of AI systems that learn from data to perform such tasks. The ML components of an AI system include the data, model, and processes for training, testing, and validation. Although AI also includes various knowledge-based approaches, the data-driven approach of ML introduces additional security challenges in training and testing (inference) phases of ML operations. These security challenges include the potential for adversarial manipulation of training data, and adversarial exploitation of model sensitivities to adversely affect the performance of ML classification and regression. AML is concerned with the design of ML algorithms that can resist security challenges, the study of the capabilities of attackers, and the understanding of attack consequences [1]. Attacks are launched by adversaries with malevolent intentions, and security of ML refers to defenses intended to prevent or mitigate the consequences of such attacks. Although ML components may also be adversely affected by various unintentional factors, such as design flaws or data biases, these factors are not intentional adversarial attacks, and they are not within the scope of security addressed by the literature on AML.

This document presents a taxonomy of concepts and defines terminology in the field of AML. The taxonomy, built on and integrating previous AML survey works, is arranged in a conceptual hierarchy that includes key types of attacks, defenses, and consequences. The terminology, arranged in an alphabetical glossary, defines key terms associated with the security of the ML components of an AI system. Taken together, the terminology and taxonomy are intended to inform future standards and best practices for assessing and managing the security of ML components, by establishing a common language and understanding of the rapidly developing AML landscape.

The literature on AML uses various terms to characterize security and assurance, including robustness and resilience. In cybersecurity more generally (NIST Glossary of Key Information Security Terms, NISTIR 7298, Revision 2), robustness refers to reliable operation of a system across a range of conditions (including attacks), and resilience refers to adaptable operations and recovery from disruptions (including attacks). Also, in cybersecurity more generally (NIST Glossary of Key Information Security Terms, NISTIR 7298, Revision 2), both robustness and resilience are gauged by risk, which is a measure of the extent to which an entity (e.g., system) is threatened by a potential circumstance or event (e.g., attack). Therefore, this general notion of risk offers a useful approach for assessing and managing the security of ML components.

As introduced in the NIST Guide for Conducting Risk Assessments (NIST 800-30, Revision 1):

> Risk assessment is one of the fundamental components of an organizational risk management process... The purpose of risk assessments is to inform decision makers and support risk responses by identifying: (i) relevant threats to organizations or threats

> directed through organizations against other organizations; (ii) vulnerabilities both internal and external to organizations; (iii) impact (i.e., harm) to organizations that may occur given the potential for threats exploiting vulnerabilities; and (iv) likelihood that harm will occur.

On that basis, a risk-based approach would begin by identifying relevant threats, vulnerabilities, and impacts. In the case of AML, threats are defined by the types of attacks and adversarial contexts in which attacks may occur; vulnerabilities are defined by the types of defenses, or lack thereof, for preventing or mitigating attacks; and impacts are defined by the consequences that result from attacks and associated defenses against those attacks. Therefore, the taxonomy of AML here is aligned with these three dimensions of AML risk assessment, namely: attacks, defenses, and consequences.

The taxonomy is presented below, by discussing key concepts in each dimension based on reviews of other taxonomies and surveys of the AML literature. In the discussion, concepts appearing in the taxonomy are written in title case italics. Because of rapid growth of concepts and methods in this field, the intent is not to be exhaustive but rather to aid readers in understanding relevant concepts pertaining to AML attacks, defenses, and consequences. Also, while the taxonomy identifies attacks, defenses, and consequences from a risk-based perspective, no attempt is made here to quantify the likelihoods and consequences that may arise from AML attacks and defenses.

The taxonomy is followed by a glossary of terminology, including a stand-alone definition for each individual term. This terminology was also extracted from existing literature and is intended to complement the taxonomy by defining additional descriptive terms that do not appear explicitly as headings in the taxonomy. Like the taxonomy, the terminology and definitions are intended not to be exhaustive but rather to aid in understanding key concepts as discussed in various other authors' reviews of the AML literature.

2 Taxonomy

The taxonomy is based on recently published papers that survey the AML literature and offer taxonomies of attacks and defenses. More than a dozen such papers, identified via keyword searches, were reviewed with the aim of identifying those themes and terms that appeared to be most prevalent among authors. Special attention was paid to papers that provided lucid explanations and recent compilations reflecting common if not consensus views across a number of authors. The primary sources used here include: Akhtar (2018) [2], Biggio (2018) [3], Chakraborty (2018) [4], Liu (2018) [5], and Papernot (2018) [6]. Additional sources used here include: Kuznetsov (2019) [7], Goodfellow (2018) [8], Yuan (2019) [9], Papernot (2017) [10], Papernot (2016) [11], Huang (2011) [1], Barreno (2010) [12], and Barreno (2006) [13].

The primary sources noted above treat topics in AML from different perspectives, with varying degrees of breadth and depth. For example, Akhtar [2], concerned with computer vision applications, addresses attacks and defenses in that domain with greater depth than the other authors noted above. Biggio [3] offers more of a historical perspective, tracing the evolution of AML with a broader focus on computer vision and cybersecurity tasks. Charkraborty [4], Liu

[5], and Papernot (2018) [6] are all concerned with cataloging attacks and defenses with an even broader focus independent of the specific area of application. Much overlap exists in these papers, with authors often citing the same sources for the topics and terms they discuss.

This NISTIR is intended to capture common aspects of these previous papers surveying the field of AML, in an integrated taxonomy adopting a risk-based perspective (see NIST Guide for Conducting Risk Assessments, NIST 800-30, Revision 1) that applies across areas of application. The highest levels of the resulting taxonomy include various aspects of *Attacks* and *Defenses*, as illustrated by Figure 1 in the context of *Training* and *Testing (Inference)* phases of the machine learning pipeline. Figure 2 organizes these and lower levels of the taxonomy in a hierarchical fashion along the three dimensions of *Attacks*, *Defenses*, and *Consequences*. The third dimension, *Consequences*, does not appear in the other taxonomies noted above and instead has been addressed by other authors as an aspect of *Attacks* dealing with the adversary's intent.

A contribution here is to address *Consequences* as a separate dimension of risk, because *Consequences* will depend on *Defenses* as well as *Attacks*, and because the actual or potential *Consequences* of *Attacks* and *Defenses* may or may not be consistent with the adversary's intent. As noted earlier, while we identify aspects of *Consequences* as well as *Attacks* and *Defenses*, we do not attempt to quantify these individual dimensions of risk or overall risk. Indeed, we expect risk will depend highly on the specific application context in which an ML component is deployed. Nevertheless, our intent is to introduce a taxonomy (and associated terminology) of AML in a manner that may support future efforts to assess and manage operational risks in practical applications of ML.

Details of each dimension in the taxonomy are summarized in sections below.

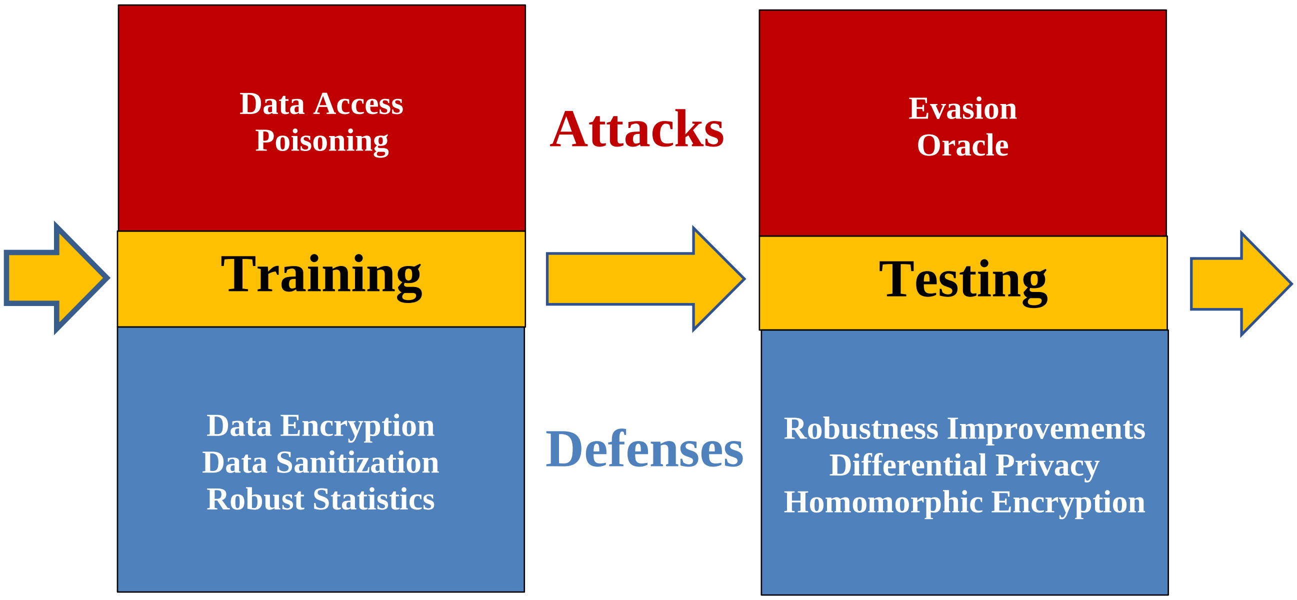

Figure 1. An illustration of example Attacks and Defenses in the Machine Learning Pipeline.

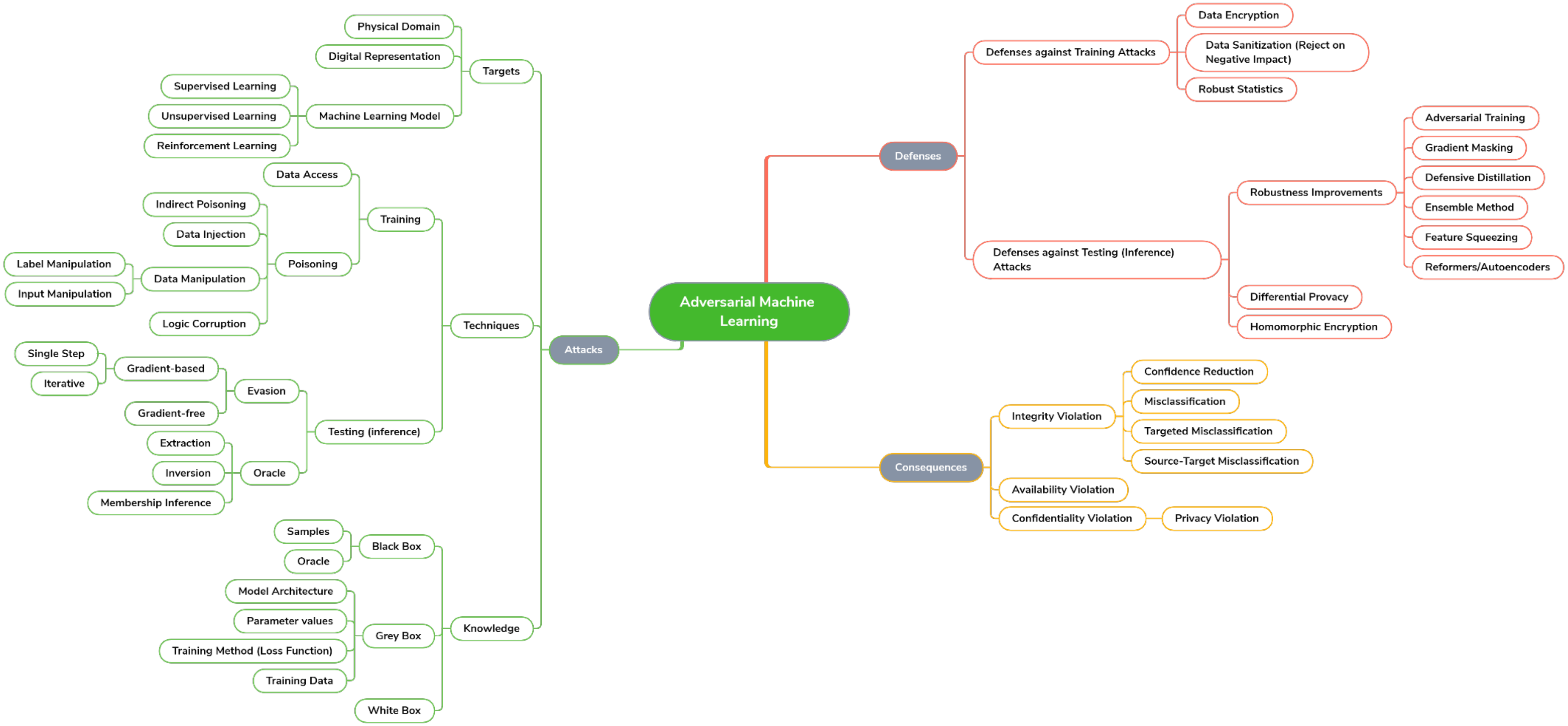

Figure 2. Taxonomy of Attacks, Defenses, and Consequences in Adversarial Machine Learning

1. Attacks
    a. Targets
        i. Physical Domain (of input sensors or output actions)
        ii. Digital Representation
        iii. Machine Learning Model
            1. Supervised Learning
            2. Unsupervised Learning
            3. Reinforcement Learning
    b. Techniques
        i. Training
            1. Data Access
            2. Poisoning
                a. Indirect Poisoning
                b. Direct Poisoning
                    i. Data Injection
                    ii. Data Manipulation
                        1. Label Manipulation
                        2. Input Manipulation
                    iii. Logic Corruption
        ii. Testing (Inference)
            1. Evasion
                a. Gradient-based
                    i. Single Step
                    ii. Iterative
                b. Gradient-free
                c. Oracle
                    i. Extraction
                    ii. Inversion
                    iii. Membership Inference
    c. Knowledge
        i. Black Box
            1. Samples
            2. Oracle
        ii. Gray Box
            1. Model Architecture
            2. Parameters Values
            3. Training Method (Loss Function)
            4. Training Data
        iii. White Box
2. Defenses
    a. Defenses Against Training Attacks
        i. Data Encryption
        ii. Data Sanitization (Reject on Negative Impact)
        iii. Robust Statistics
    b. Defenses Against Testing (Inference) Attacks

        i. Robustness Improvements
            1. Adversarial Training
            2. Gradient Masking
            3. Defensive Distillation
            4. Ensemble Method
            5. Feature Squeezing
            6. Reformers/Autoencoders
        ii. Differential Privacy
        iii. Homomorphic Encryption
3. Consequences
    a. Integrity Violation
        i. Confidence Reduction
        ii. Misclassification
        iii. Targeted Misclassification
        iv. Source-Target Misclassification
    b. Availability Violation
    c. Confidentiality Violation
        i. Privacy Violation

2.1 Attacks

ML components may be *Targets* of *Attacks* by adversaries using various *Techniques* and *Knowledge* about the systems.

2.1.1 Targets

The *Targets* of *Attacks* are defined by stages in the ML pipeline, including the *Physical Domain* of input sensors, the *Digital Representation* for pre-processing, the *Machine Learning Model* itself, or the *Physical Domain* of output actions. The types of methods generating a *Machine Learning Model* include *Supervised Learning*, *Unsupervised Learning*, and *Reinforcement Learning*. In *Supervised Learning*, training data are provided in the form of inputs labeled with corresponding outputs, and the model learns a mapping between inputs and outputs. The learning task is referred to as classification when the outputs take on categorical values, and regression when the outputs take on numerical values. In *Unsupervised Learning*, training data are unlabeled inputs, and the model learns an underlying structure of the data. For example, the model may perform clustering of inputs according to some similarity metric, or dimensionality reduction to project data into lower dimensional subspaces. In *Reinforcement Learning*, a reward-based policy for acting in an environment is learned from training data represented as sequences of actions, observations, and rewards. In some applications, *Reinforcement Learning* may be combined with *Supervised Learning* and *Unsupervised Learning*. Although all three types of systems may be *Targets* of *Attacks*, most research in AML has focused on *Supervised Learning* systems, typically as applied to image classification tasks. However, algorithms developed to craft adversarial examples for classification are equally applicable to reinforcement learning [6].

2.1.2 Techniques

Adversarial *Techniques* used for launching *Attacks* against *Targets* may apply to the *Training* or

Testing (Inference) phases of system operation. *Attacks* in the *Training* phase attempt to acquire or influence the training data or model itself. In *Data Access Attacks*, some or all of the training data is accessed and can be used to create a substitute model. This substitute model can then be used to test the effectiveness of potential inputs before submitting them as *Attacks* in the *Testing (Inference)* phase of operation. In *Poisoning*, also known as *Causative Attacks*, the data or model are altered indirectly or directly. In *Indirect Poisoning*, adversaries without access to pre-processed data used by the target model must instead poison the data before pre-processing. In direct poisoning, the data are altered by *Data Injection* or *Data Manipulation*, or the model is altered directly by *Logic Corruption*. In *Data Injection*, adversarial inputs are inserted into the original training data, thereby changing the underlying data distribution without changing the features or labels of the original training data. Injected adversarial samples can be optimized by linear programming methods that shift the decision boundary of a centroid model (in *Unsupervised Learning*), or by gradient ascent on the test error of the model to degrade classification accuracy (in *Supervised Learning*). *Data Manipulation* involves adversarial modification of output labels (*Label Manipulation*) and input data (*Input Manipulation*) of the original training data. *Logic Corruption* is accomplished by an adversary who can tamper with the ML algorithm and thereby alter the learning process and model itself.

Attacks in the *Testing (Inference)* phase, also known as *Exploratory Attacks*, do not tamper with the target model or the data used in training. Instead these *Attacks* generate adversarial examples as inputs that are able to evade proper output classification by the model, in *Evasion Attacks*, or collect and infer information about the model or training data, in *Oracle Attacks*.

In *Evasion Attacks*, the adversary solves a constrained optimization problem to find a small input perturbation that causes a large change in the loss function and results in output misclassification. This typically involves *Gradient-based* search algorithms such as Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS), Fast Gradient Sign Method (FGSM), or Jacobian-based Saliency Map Attack (JSMA). L-BFGS was the first algorithm used to generate misclassifications by a computer vision system model using input perturbations that were imperceptible to human observers. FGSM improves the computational efficiency of gradient ascent, in a *Single Step* approach that eliminates iterations required to obtain a perturbation that will cause a large change in the loss function. Compared to FGSM, JSMA is an *Iterative Algorithm* that provides more fine-grained control of perturbed features and thereby can generate more convincing adversarial examples, albeit at increased computational cost. These and other algorithms for *Evasion Attacks* require knowledge of the model, or a substitute model, in order to compute gradients in the loss function across input-output pairings. Besides L-BFGS, FGSM, and JSMA, many other techniques with similar operating principles have been developed to generate adversarial examples [2] [4] [9], one of which is depicted in Figure 3 [14]. *Gradient-free* attacks [15] have also been developed, but they typically require access to model confidence values in order to be effective.

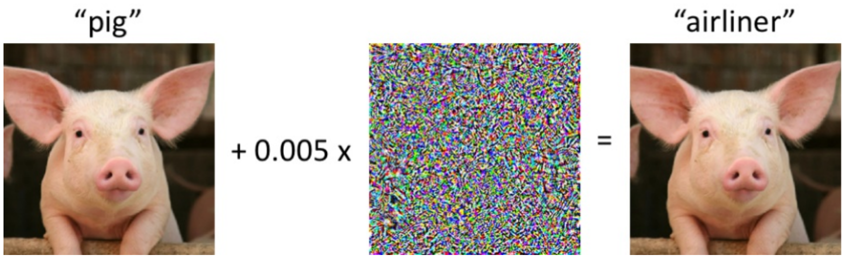

Figure 3. Example of adversarial perturbation used to evade classifiers [14].

In *Oracle Attacks*, an adversary uses an Application Programming Interface to present the model with inputs and to observe the model's outputs. Even when the adversary has no direct knowledge of the model itself, the input-output pairings obtained from *Oracle Attacks* can be used to train a substitute model that operates much like the target model, due to the transferability property exhibited by many model architectures. This substitute model, in turn, can then be used to generate adversarial examples for use in *Evasion Attacks* against the target model. *Oracle Attacks* include *Extraction Attacks*, *Inversion Attacks*, and *Membership Inference Attacks*. These attacks collect information such as output and confidence values, to infer parameters or characteristics of the model or data. In *Extraction Attacks*, an adversary extracts the parameters or structure of the model from observations of the model's predictions, typically including probabilities returned for each class. In the case of *Inversion Attacks*, the inferred characteristics may allow the adversary to reconstruct data used to train the model, including personal information that violates the privacy of an individual. In a *Membership Inference Attack*, the adversary uses returns from queries of the target model to determine whether specific data points belong to the same distribution as the training dataset, by exploiting differences in the model's confidence on points that were or were not seen during training.

2.1.3 Knowledge

Besides *Techniques* used to launch *Attacks* against *Targets*, threats to ML components also depend on the adversary's *Knowledge* about the target model. In *Black Box Attacks*, the adversary has no knowledge about the model except input-output *Samples* of training data or input-output pairings obtained using the target model as an *Oracle*. In *Gray Box Attacks*, the adversary has partial information about the model, which may include the *Model Architecture*, *Parameter Values*, *Training Method (Loss Function)*, or *Training Data*. In *White Box Attacks*, the adversary has complete knowledge of the model including architecture, parameters, methods, and data. Even when an adversary does not have the complete knowledge needed for a *White Box Attack*, *Data Access* or *Oracle Attacks* that produce input-output pairings can be used to train a substitute model, which operates much like the actual model due to the transferability property exhibited by many model architectures. This substitute model can then be used as a *White Box* to generate adversarial examples for use in *Evasion Attacks*.

2.2 Defenses

Defenses can be characterized by whether they apply to *Attacks* launched against the *Training* or *Testing (Inference)* phases of system operation. In both cases, defensive methods often can incur performance overhead as well as have a detrimental effect on model accuracy [4].

Defenses Against Training Attacks involving *Data Access* include traditional access control measures such as *Data Encryption*. *Defenses* against *Poisoning Attacks* include *Data Sanitization* and *Robust Statistics*. In *Data Sanitization*, adversarial examples are identified by testing the impacts of examples on classification performance. Examples that cause high error rates in classification are then removed from the training set, in an approach known as *Reject on Negative Impact*. Rather than attempting to detect poisoned data, *Robust Statistics* use constraints and regularization techniques to reduce potential distortions of the learning model caused by poisoned data.

Defenses Against Testing (Inference) Attacks include various model *Robustness Improvements*, including *Adversarial Training*, *Gradient Masking*, *Defensive Distillation*, *Ensemble Methods*, *Feature Squeezing*, and *Reformers/Autoencoders*. Although used as *Defenses* against *Attacks* made in the *Testing (Inference)* phase, these *Defenses* are deployed by the defender in the *Training* phase that precedes *Testing (Inference)*. In *Adversarial Training*, inputs containing adversarial perturbations but with correct output labels are injected into the training data in order to minimize classification errors caused by adversarial examples. *Gradient Masking* reduces the model's sensitivity to small perturbations in inputs by computing first order derivatives of the model with respect to its inputs and minimizing these derivatives during the learning phase. A similar idea motivates *Defensive Distillation*, where a target model is used to train a smaller model that exhibits a smoother output surface, and *Ensemble Methods*, where multiple classifiers are trained together and combined to improve robustness. Similarly, *Feature Squeezing*, shown in Figure 4, uses smoothing transformations of input features in an attempt to undo adversarial perturbations [16]. *Reformers* take a given input and push it toward the closest example in the training set, typically using neural networks called *Autoencoders*, to counter adversarial perturbations.

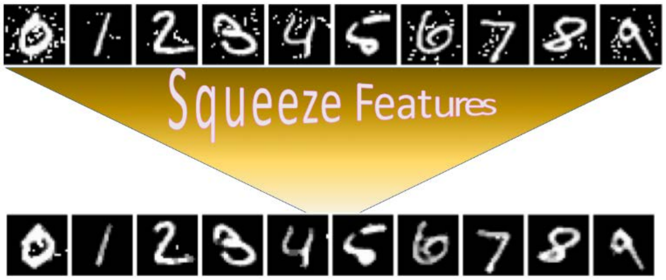

Figure 4. An example of Feature Squeezing, which smooths inputs to remove adversarial inputs [16].

It is important to acknowledge that the adversary may defeat various Robustness Improvement Defenses by launching Data Access or Oracle Attacks to obtain input-output pairings. These pairings can be subsequently used to train a substitute model that does not mask gradients or smooth outputs like the target model. The substitute model can then be used as a White Box to craft adversarial examples, by exploiting the transferability property of ML-trained models, so it can be difficult to defend against Evasion Attacks by an adversary capable of creating a substitute model.

Besides the *Robustness Improvements* noted above, *Defenses Against Testing (Inference) Attacks* also include randomization mechanisms applied to training data or model outputs to provide *Differential Privacy* guarantees. *Differential Privacy* formulates privacy as a property satisfied by a randomization mechanism on pairs of adjacent datasets. Ultimately, the *Differential Privacy* property ensures that model outputs do not reveal any additional information about an individual record included in the training data. However, there is an inherent performance tradeoff because a model's prediction accuracy is degraded by the randomization mechanisms used to achieve *Differential Privacy*. An alternative approach is *Homomorphic Encryption*, which encrypts data in a form that a neural network can process without decrypting the data. This protects the privacy of each individual input but introduces computational performance overhead and limits the set of arithmetic operations to those supported by *Homomorphic Encryption*.

2.3 Consequences

The *Consequences* of *Attacks* against *Targets* depend on implemented *Defenses*. For a given combination of *Attack* (including *Target*, *Technique*, and *Knowledge*) and *Defense(s)*, the *Consequences* can be characterized categorically as *Violations* of *Integrity*, *Availability*, *Confidentiality*, or *Privacy*. Within each category, varying levels of severity may also be used to measure the violation of security.

In *Integrity Violations*, the inference process is undermined, resulting in *Confidence Reduction* or *Misclassification* to any class different from the original class. More specific misclassifications include *Targeted Misclassification* of inputs to a specific target output class and *Source-Target Misclassification* of a specific input to a specific target output class. In *Unsupervised Learning*, an *Integrity Violation* may produce a meaningless representation of the input in an unsupervised feature extractor. In *Reinforcement Learning*, an *Integrity Violation* may cause the learning agent to act unintelligently or with degraded performance in its environment.

Availability Violations induce reductions in quality (such as inference speed) or access (denial of service) to the point of rendering the ML component unavailable to users. Although *Availability Violations* may involve *Confidence Reductions* or *Misclassifications* similar to those of *Integrity Violations*, the difference is that *Availability Violations* result in behaviors such as unacceptable speed or denial of access that render a model's output or action unusable.

Confidentiality Violations occur when an adversary extracts or infers usable information about the model and data. *Attacks* on confidential information about the model include an *Extraction Attack* that reveals model architecture or parameters, or an *Oracle Attack* that enables the adversary to construct a substitute model. Attacks that reveal confidential information about the data include an *Inversion Attack* whereby an adversary exploits the target model to recover missing data using partially known inputs, or a *Membership Inference Attack* whereby an adversary performs a membership test to determine if an individual was included in the dataset used to train the target model.

Privacy Violations are a specific class of *Confidentiality Violation* in which the adversary obtains personal information about one or more individual and legitimate model inputs, either included in the training data or not. An example would be when an adversary acquires or extracts an individual's medical records in violation of privacy policies.

3 Terminology

As a complement to the taxonomy discussed above, this section presents a glossary of terminology with a stand-alone definition for each term.

Similar to the taxonomy, the terminology is based on recently published papers that survey the AML literature as well as papers that address recent advances in the field. These papers were reviewed with the aim of identifying those themes and terms that appeared to be most prevalent among authors. The primary sources used here include: Akhtar (2018) [2], Biggio (2018) [3], Chakraborty (2018) [4], Liu (2018) [5], and Papernot (2018) [6]. Additional sources used here include: Kuznetsov (2019) [7], Goodfellow (2018) [8], Yuan (2019) [9], Papernot (2017) [10], Papernot (2016) [11], Huang (2011) [1], Barreno (2010) [12], and Barreno (2006) [13]. Terminology definitions were constructed from the identified themes and terms.

The field of AI Security is currently heavily centered around AML, and much of the terminology draws from the fields of ML. The goal and contribution of this NISTIR terminology is to aggregate those terms that are in common usage in AML and use the sources to compile common, standardized definitions. The guidelines for selecting terms for inclusion here are that the terms are not general ML (e.g., deep learning) terms that are likely already defined in that more general fields. Also excluded are terms that are specifically named and published algorithms. In case of varying definitions, definitions were prioritized based on recency, generality, and most common usage in source surveys. The references provided indicate one or more possible sources of relevant information or the stated definition. They are not intended to indicate specific endorsement or to assign originator credit.

Table 1. Terminology. This table lists terms, synonyms for these terms, definitions, and references for these definitions.

ID	Term	Synonym	Assigned Definition	Reference
1	**Adversarial capabilities**		The various actions, information, techniques or attack vectors available to an attacker on a threat surface.	[6]
2	**Activation maximization**		The synthetization of inputs that activate specific neurons in a neural network to produce synthetic inputs that are human-interpretable.	[6]
3	**Adversarial example transferability**		The property that adversarial examples crafted to be misclassified by a model are likely to be misclassified by a different model.	[6]

4	**Adversarial example**		ML input sample formed by applying a small but intentionally worst-case perturbation (see adversarial perturbation) to a clean example, such that the perturbed input causes a learned model to output an incorrect answer.	[3], [2]
5	**Adversarial perturbation**		The noise added to an input sample to make it an adversarial example.	[2]
6	**Adversarial training**		Defensive method to increase model robustness by injecting adversarial examples into the training set.	[4]
7	**Adversary**		The agent who conducts or intends to conduct detrimental activities, perhaps by creating an adversarial example.	[2], [17]
8	**Attack**		Action targeting a learning system to cause malfunction.	[13]
9	**Attack detection**		The action of differentiating between anomalous and normal behavior, or between an adversarial example and a benign example.	[6]
10	**Attack detector**		A mechanism to (only) detect if a sample is an adversarial.	[2]
11	**Autoencoder attack**		A perturbation attack on autoencoders that leads the autoencoder to reconstruct a completely different image.	[2]
12	**Auxiliary model**	Substitute or Surrogate model	An attacker's model trained to approximate the decision boundary of the target model. Useful for testing attacks offline.	[2], [4], [6]

13	**Availability violation**		A compromise of the normal system functionalities available to legitimate users, such as accuracy, quality, or access, resulting in inaccessible or unusable model output.	[3], [6]
14	**Black-box attack**	Zero-knowledge attack	Attack that assumes no knowledge about the model under attack. The adversary may use context or historical information to infer model vulnerability. The attacker may probe the system to inform system vulnerabilities.	[2], [6], [4]
15	**Causative attack**	Poisoning attack	See "Poisoning Attack."	[13]
16	**Confidence reduction**		Reducing the confidence of prediction for the target model. For example, a legitimate image of a 'stop' sign can be predicted with a lower confidence having a lesser probability of class membership.	[4]
17	**Confidentiality attack**		An attack in which the adversarial goal is to reveal evidence of a model's characteristics or information about its training data.	[6], [3]
18	**Data sanitization**		Defensive method that identifies and treats manipulated samples as outliers in the training data, to be detected and removed.	[18], [3]
19	**Dataset modification**		Altering the training data directly, in contrast to injection.	[6]
20	**Deep Contractive Network**		An ML technique in which, for defensive purposes, a smoothness penalty is applied to reduce susceptibility to adversarial examples. It penalizes output	[2], [6]

			variation with respect to input variation to increase the variation needed to produce adversarial examples.	
21	**Defensive distillation**	Distillation	A procedure to train deep neural network (DNN)-based classifier models that are more robust to perturbations. Distillation extracts additional knowledge about training points as probability vectors produced by a DNN, which is fed back into the training regimen. Distillation generates smoother classifier models by reducing their sensitivity to input perturbations. These smoother DNN classifiers are found to be more resilient to adversarial samples and have improved class generalizability properties. A type of gradient masking.	[19], [4]
22	**Dense evasion attack**	L2-norm attack	Evasion (L2-norm) attack where the cost of modifying features is proportional to the distance between the original and modified sample in Euclidean space. The attacker will prefer to make small changes to many or all features.	[20]
23	**Differential privacy**		A mathematical formulation that defines the privacy provided by an ML model as the property that a learning algorithm's output will not differ statistically by the change of a single training example. This formulation is leveraged by multiple defenses that aim to protect data privacy.	[6], [5]

24	**Disinformation technique**		Altering data seen by the adversary with the goal of confusing the adversary's estimate of the learner's state.	[13]
25	**Distinguishability measure**		A measurement of classifier robustness that describes the difference between classes of a dataset. Distinguishability is the distance between the means of two classes for linear classifiers and the distance between the matrices of second order moments for non-linear classifiers.	[2]
26	**Distribution drift**		A situation in which the training and test input distributions differ.	[6]
27	**Enchanting attack**		An attack on deep reinforcement learning in which the adversary lures the attacked system to a designated target state by integrating a generative model and a planning algorithm. The generative model is used for predicting the future states of the agent, whereas the planning algorithm generates the actions for luring it.	[2]
28	**Ensemble learning or method**		A classification method using multiple classifiers to enhance robustness including against evasion attacks.	[3], [21]
29	**Error specificity**		Describes the misclassification goal of an attacker: if the attacker aims to have a sample misclassified as a specific class, specificity is specific (targeted attack); if the attacker aims for any misclassification, specificity is generic (non-targeted attack).	[3]

30	**Error-generic evasion attack**		The attacker is interested in causing a misclassification of a test sample, regardless of the output class predicted by the classifier.	[3]
31	**Error-generic poisoning attack**		The attacker, using training set poisoning, aims to cause a denial of service, by inducing as many misclassifications as possible (regardless of the classes in which they occur).	[3]
32	**Error-specific evasion attack**		The attacker aims to mislead classification of a test sample, such that the adversarial samples are misclassified as a specific class.	[3]
33	**Error-specific poisoning attack**		The attacker, using training set poisoning, aims to cause specific types of misclassifications.	[3]
34	**Evasion attack**		The attacker manipulates input samples to evade (cause a misclassification) a trained classifier at test time.	[3]
35	**Explainability**		The ability to provide a human-interpretable explanation for an ML prediction and produce insights about the causes of decisions, potentially to line up with human reasoning.	[22]
36	**Exploratory attack**		The attacker manipulates only test data. Aims to cause misclassification with respect to adversarial samples (evasion) or to uncover sensitive information from training data and learning models (oracle).	[3], [5]
37	**Fast Gradient Sign Method**		An efficient method for computing an adversarial image perturbation, using the gradient	[2]

	(FGSM)		of the cost function. The image is perturbed to increase the loss of the classifier on the resulting image.	
38	**Fast-flipping attribute technique (attack)**		An attack on facial recognition which imperceptibly modifies a single attribute to cause the face to be wrongly classified. Adversarial images are generated by flipping the binary decision of a deep neural network.	[2], [23]
39	**Foveation Based Defense**		An ML technique in which neural networks are applied to segments of images to improve robustness to adversarial patterns in the images.	[2]
40	**Generative adversarial network**		An ML technique which increases the effectiveness of a model generator by training it in the presence of an adversary—a discriminator which seeks to differentiate between real data and generated data. The effectiveness of the generator is measured by the error rate of the discriminator. Used in the generation of training data in an autoencoder attack or as a defense to train a more robust classifier.	[2]
41	**Generative model**		An ML model trained with the goal of generating new data points. The model takes a training set, consisting of samples drawn from a distribution, and learns to represent an estimate of that distribution. As an attack, the generative model is trained to generate candidate adversarial samples.	[2], [8], [5]

42	**Generic specificity**		Describes the goal of an attack as misclassifying a sample as any of the classes different from its true class.	[3]
43	**Gradient ascent**		An iterative algorithm used to find a minimum of a function. Identifies the optimal adversarial inputs corresponding to local maxima in the test error of the model. Operates by calculating the gradient of objective functions that measure effectiveness.	[6], [4], [5]
44	**Gradient masking**		An ML technique in which gradients are minimized to reduce the model's sensitivity to adversarial examples. Hides the gradient direction used to craft adversarial examples.	[6], [3]
45	**Gray-box attack (grey-box attack)**	Limited knowledge attack	Attack which assumes partial knowledge about the model under attack (e.g., type of features, or type of training data).	[2], [3]
46	**Homomorphic encryption**		A technique in which encrypted data can be processed by a neural network without decryption, allowing for data protection and improving data privacy when processed by an ML algorithm.	[6], [5]
47	**Image perturbation**		A change or transformation to an image, often to cause a misclassification.	[2]
48	**Indiscriminate attack**		An attack that aims to cause misclassification of any sample to target any system user or protected service.	[3], [5]
49	**Inference**		The stage of ML in which a model is applied to a task. For example, a classifier model	[6]

			produces the classification of a test sample.	
50	**Injection (data injection) attack**		The insertion of adversarial inputs into the existing training data.	[6]
51	**Input manipulation attack**		A threat model that assumes the adversary can corrupt the input features of training samples or training sample labels.	[6]
52	**Integrity violation**		To induce a particular output or behavior of the adversary's choosing. Compare against Confidentiality and Availability violations.	[6]
53	**Jacobian-based Saliency Map Attack (JSMA)**		An attack that makes optimal miniscule changes to input data until the classifier is fooled or a maximum number of changes is met.	[2], [6]
54	**L2-norm attack**	Dense evasion attack	See “Dense evasion attack.”	
55	**Label manipulation attack**		An attack in which the adversary corrupts the labels of training data.	[6], [4]
56	**Label smoothing defense**		A defense mechanism in which labels are changed from classes to real numbers, allowing for classification outside of the strict class labels.	[6], [4]
57	**Limited-knowledge attack**	Gray-box, or semi-black box	See “Gray-box attack.”	
58	**Linearity hypothesis**		The hypothesis that designs of DNNs that intentionally encourage linear behavior for computation efficiency, make them susceptible to cheaper	[2]

			adversarial perturbations.	
59	**Logic corruption attack**		An attack on an ML model in which the learning algorithm or logic itself is tampered with.	[6], [4]
60	**Membership attack**		An attack that targets the information of whether or not a given data point was part of the training dataset or part of the same distribution as the training dataset.	[6], [4]
61	**Misclassification attack**		Attack to alter the output classification of an input example to any class different from its true class. For example, a legitimate image of a ‘stop’ sign will be predicted as any other class different from the class of stop sign.	[4]
62	**Model extraction attack**		An exploratory attack that aims to discover the structure or parameters of the model by observing its predictions.	[6], [4]
63	**Model inversion attack**		An oracle attack that aims to discover training data and other sensitive data through knowledge of the model and auxiliary data.	[6]
64	**Non-targeted attack**	Untargeted attack	An attack that causes any misclassification as opposed to causing classification into a specific (incorrect) class. The predicted label of the adversarial example is irrelevant, as long as it is not the correct label. See also “Error specificity.”	[2], [6]

65	**Obfuscation attack**		An attack against a targeted cluster of samples that attempts to generate a blend of adversarial samples and normal ones from other clusters without altering the clustering results of these normal samples, resulting in a set of stealthy adversarial samples.	[5]
66	**Obfuscation defense**		A defense mechanism in which details of the model or training data are kept secret.	[6]
67	**One Pixel Attack**		An (evasion) attack that alters a single pixel in an image to cause a misclassification.	[2]
68	**One-shot/one-step method**		Generates an adversarial perturbation by performing a single step computation, e.g. computing gradient of model loss once. The opposite are iterative methods that perform the same computation multiple times to get a single perturbation. The latter are often computationally expensive.	[2]
69	**Oracle attack**		An attack in which an adversary is able to craft inputs and receive outputs to the attacked model, in an attempt to learn information about the model and craft better attacks.	[6]
70	**Output randomization**		A defense randomizing the classifier’s output to give imperfect feedback to the attacker.	[3]
71	**Outsiders**		External users or adversaries that may be able to influence a system, not including enterprise users (consumers).	[6]

72	**Perceptual distance**		Measures how similar two images are in a way that coincides with human judgment.	[24]
73	**Perfect-knowledge attack**	White-box attack	Attack that exploits model internal information. It assumes complete knowledge of the targeted model, including its parameter values, architecture, training method, and in some cases its training data as well.	[3]
74	**Poisoning attack**		Aims to increase the number of misclassified samples at test time by injecting a small fraction of carefully designed adversarial samples into the training data. Indirect poisoning manipulates data before any preprocessing, while direct poisoning the data are altered by Data Injection or Data Manipulation, or the model is altered directly by Logic Corruption. Also known as a contamination of the training data. Alternately, also includes tampering with the ML algorithm itself, to compromise the whole learning process.	[4], [5]
75	**Privacy preserving model**		A model that does not reveal personal details that may be included in its training data.	[6]
76	**Privacy violation**		Revealing personal information about an individual included in the training data.	[6]
77	**Quantitative input influence**		A measurement of the influence of certain inputs on model output.	[6]
78	**Quasi-imperceptible perturbation**		Perturbation that impairs images very slightly for human perception.	[2]

79	**Randomization defense**		A defense mechanism that adds random noise to the training data, the model training cost function, the learned parameters, or model output to preserve privacy.	[3], [6]
80	**Reactive defenses**		Defenses that aim to counter past attacks, for example, by analysis of the target classifier, by timely detection of novel attacks, by frequent classifier retraining, or by verification of consistency of classifier decisions.	[3], [5]
81	**Real-world attacks**		Attacks successfully executed on existing systems.	[2]
82	**Regularization**		A mechanism at training to improve generalizability of the model. It reduces model sensitivity or complexity, with the intent of limiting exploitability.	[6], [2]
83	**Resilience**		“The ability to prepare for and adapt to changing conditions and withstand and recover rapidly from disruptions. Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents.” The ability of a system to adapt to and recover from adverse conditions.	[25]
84	**Robust learning**		Learning algorithms based on robust statistics that are intrinsically less sensitive to outlying training samples.	[3]
85	**Robust optimization**		Formulates adversarial learning as a mini- max problem in which the inner problem maximizes the training loss by manipulating the training points under worst-case,	[3]

			bounded perturbations, while the outer problem trains the learning algorithm to minimize the corresponding worst-case training loss.	
86	**Robustness**		The ability of an ML model/algorithm to maintain correct and reliable performance under different conditions (e.g., unseen, noisy, or adversarially manipulated data)	[2], [25]
87	**Sample rejection defense**		Defensive mechanism detecting and rejecting samples that are sufficiently far (as measured by a distance metric) from the training data in feature space.	[3]
88	**Security evaluation curve**		Shows the extent to which the performance of a learning algorithm drops gracefully under attacks of increasing strength.	[3]
89	**Source-target misclassification attack**		An adversarial attempt to force the output of classification for a specific input to be a particular target class. For example, the input image of ‘stop’ sign will be predicted as a ‘Speed Limit’ sign by the classification model.	[4]
90	**Sparse evasion attack**		Attack (using L1-norm) where cost depends on the number of modified features, and attacker aims to minimize the number of modified features.	[20]
91	**Specific error**		Describes the goal of an attack as misclassifying a sample as a specific class.	[3]
92	**Strategically-timed attack**		An attack on reinforcement learning in which the adversary attacks the model in a small subset of time steps to affect the	[2]

			model's behavior without detection.	
93	**Substitute model or network**	Surrogate or Auxiliary model	See “Auxiliary model.”	
94	**Surrogate model**	Substitute or Auxiliary model	See “Auxiliary model.”	
95	**Targeted misclassification attack**		The adversary tries to produce inputs that force the output of the classification model to be a specific target class. For example, any input image to the classification model will be predicted as a class of images having a ‘Speed Limit’ sign. See “Error specificity.”	[4]
96	**Threat model**		Adversarial goals, knowledge, and capabilities that a system is designed to defend against.	[6], [3], [4]
97	**Training data extraction attack**		An attack in which the goal is to discover parts or all of the training data.	[6]
98	**Transferability of example**		The ability of an adversarial example to remain effective even for the models other than the one used to generate it.	[2], [6]
99	**Transparency**		Understanding the working logic of the model.	[26]
100	**Trust model**		A description of the level of trust assigned to various actors in a system deployment. Actors include data owners, system providers, service consumers, and outsiders who access or influence the system.	[6]

101	**Universal (Adversarial) perturbation**		Perturbation able to fool a given model on ‘any’ image with high probability. Note that, universality refers to the property of a perturbation being ‘image-agnostic’ as opposed to having good transferability.	[2]
102	**Untargeted attack**	Non-targeted attack	See “Non-targeted attack.”	
103	**White-box attack**	Perfect knowledge attack	See “Perfect knowledge attack.”	[2], [6], [4]
104	**Zero-knowledge attack**	Black-box attack	See “Black-box attack”.	[3]

537 References

[1] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein and J. D. Tygar, "Adversarial Machine Learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence,* New York, NY, USA, 2011.

[2] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access,* vol. 6, pp. 14410-14430, 2018.

[3] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition,* vol. 84, pp. 317-331, 2018.

[4] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay and D. Mukhopadhyay, "Adversarial Attacks and Defences: A Survey," 28 9 2018.

[5] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu and V. C. M. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE access,* vol. 6, pp. 12103-12117, 2018.

[6] N. Papernot, P. McDaniel, A. Sinha and M. P. Wellman, "SoK: Security and privacy in machine learning," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P),* 2018.

[7] P. Kuznetsov, R. Edmunds, T. Xiao, H. Iqbal, R. Puri, N. Golmant and S. Shih, "Adversarial Machine Learning," in *Artificial Intelligence Safety and Security*, Chapman and Hall/CRC, 2018, pp. 235-248.

[8] I. Goodfellow, P. McDaniel and N. Papernot, "Making machine learning robust against adversarial inputs," *Communications of the ACM,* vol. 61, pp. 56-66, 2018.

[9] X. Yuan, P. He, Q. Zhu and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems,* 2019.

[10] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security,* 2017.

[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik and A. Swami, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, 2016.

[12] M. Barreno, B. Nelson, A. D. Joseph and J. D. Tygar, "The security of machine learning," *Machine Learning,* vol. 81, pp. 121-148, 2010.

[13] M. Barreno, B. Nelson, R. Sears, A. D. Joseph and J. D. Tygar, "Can machine learning be secure?," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security,* 2006.

[14] A. Mądry and L. Schmidt, "A Brief Introduction to Adversarial Examples," Gradient Science, [Online]. Available: http://gradientscience.org/intro_adversarial/. [Accessed 19 July 2019].

[15] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry and A. Kurakin, "On Evaluating Adversarial Robustness," 18 2 2019.

[16] W. Xu, D. Evans and Y. Qi, "Is Robust Machine Learning Possible?," EvadeML (University of Virginia), [Online]. Available: https://evademl.org/. [Accessed 19 July 2019].

[17] G. Stoneburner, A. Y. Goguen and A. Feringa, "SP 800-30 Rev.1 Guide for Conducting Risk Assessments," National Institute of Standards & Technology, Gaithersburg, MD, United States, 2012.

[18] J. Steinhardt, P. W. W. Koh and P. S. Liang, "Certified defenses for data poisoning attacks," in *Advances in neural information processing systems*, 2017.

[19] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016.

[20] A. Demontis, P. Russu, B. Biggio, G. Fumera and F. Roli, "On security and sparsity of linear classifiers for adversarial settings," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2016.

[21] B. Biggio, G. Fumera and F. Roli, "Multiple classifier systems for robust classifier design in adversarial environments," *International Journal of Machine Learning and Cybernetics*, vol. 1, pp. 27-41, 2010.

[22] F. Doshi-Velez and M. Kortz, "Accountability of AI Under the Law:," 21 November 2017. [Online]. Available: https://arxiv.org/pdf/1711.01134.pdf. [Accessed 17 September 2019].

[23] A. Rozsa, M. Gunther and T. E. Boult, "Towards robust deep neural networks with BANG," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[24] H. Zhang, I. Goodfellow, D. Metaxas and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[25] C. National Security Systems Glossary Working Group, "Committee on National Security Systems (CNSS) Glossary," Gaithersburg, 2010.

[26] F. K. Dosilovic, M. Brcic and N. Hlupic, "Explainable artificial intelligence: A survey," in *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2018.

[27] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[28] H. Stern, J. Mason and M. Shepherd, "A linguistics-based attack on personalised statistical e-mail classifiers," *See http://www. cs. dal. ca/research/techreports/2004/CS-2004-06. shtml*, 2004.

[29] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[30] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter and L. Kagal, "Explaining Explanations: An Overview of Interpretability of Machine Learning," in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 2018.