AI-ASsisted cybersecurity platform empowering SMEs to defend against adversarial AI attacks

WP3 – AIAS Adversarial AI Engine and Deception
D3.2 Taxonomy of Adversarial AI attacks

Editors	UPRC
Authors	Athanasia Sampazioti (UPRC), Ignacio Lacalle (UPV), Giorgio Bernardinetti (CNIT), Iman Hasnaouia (UMA), Anastasia Tsiota (FOGUS), Andrea Paci (CNIT), Pasquale Caporaso (CNIT), Luis Miguel Campos (PDM), Athanasios Kalogeras (ISI), Ilias Politis (ISI)
Dissemination Level	PU
Type	R
Version	2

Project Profile	
Contract Number	101131292
Acronym	AIAS
Title	AI-ASsisted cybersecurity platform empowering SMEs to defend against adversarial AI attacks
Start Date	Jan 1st, 2024
Duration	48 Months

Partners

	University of Piraeus Research Center	EL
	BEIA CONSULT INTERNATIONAL SRL	RO
	UNIVERSIDAD DE MALAGA	ES
	K3Y	BG
	ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS	EL
	SUITE5 DATA INTELLIGENCE SOLUTIONS LIMITED	CY
	CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI	IT
	FOGUS INNOVATIONS & SERVICES P.C	EL
	UNIVERSITAT POLITECNICA DE VALENCIA	ES
	PDM E FC PROJECTO DESENVOLVIMENTO MANUTENCAO FORMACAO E CONSULTADORIALDA	PT

Document History

VERSIONS

Table 1 Document history

Version	Date	Author	Remarks
V0.1	30/03/2025	UPRC	Table of contents
V0.5	10/04/2025	ALL	Introduction section, Executive summary, description of attack categories
V0.7	26/04/2025	ALL	Inclusion of attacks in each category
V0.8	15/05/2025	ALL	Addition of more attacks
V0.9	22/05/2025	ALL	Conclusion and impact assessment
V0.95	30/05/2025	ALL	Description of taxonomies found in the literature and comparison with the AIAS taxonomy
V1.0	13/06/2025	ALL	Initial review version of the deliverable.
V1.2	30/9/2025	ALL	Threat model
V1.2	31/10/2025	ALL	Motivation
V1.3	1/12/2025	ALL	Discussion (refined)
V1.4	12/12/2025	ALL	Revision
V1.5	19/12/2025	ALL	Refined version
V2	31/12/2025	UPRC	Final version

AIAS message

Executive Summary

The purpose of this deliverable is to document the work carried out in the context of “*D3.2 Taxonomy of Adversarial AI attacks*” of Task 3.3 “*Taxonomy of AI adversarial attacks*” and its results until month 24. The results of this task as of the date this report is written are:

- A survey of adversarial AI attacks in a variety of application domains that are available in the literature.
- A classification of the AI attacks accumulated during the aforementioned survey using a large scale taxonomy.
- This classification was done using unique traits that include but not limited to, the attacker’s knowledge of the AI/ML model, and the time the attack occurs and the application domain of the attack.

In addition to documenting the results, this deliverable describes in detail the methodology that followed to perform this survey. Furthermore, the relation of D3.2 to other deliverables and tasks is given as defined in the project’s Grant Agreement.

Table of Contents

Table of Figures

Table of Tables

Table of Equations

Table 2. Abbreviation Table

Abbreviation	**Description**
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASR	Attack Success Rate
BIM	Basic Iterative Method
CIA	Confidentiality Integrity Availability
CLM	Code Language Model
CNN	Convolutional Neural Networks
CV	Computer Vision
CW	Carlini & Wagner
DL	Deep Learning
DNN	Deep Neural Networks
DoS	Denial of Service
FGM	Fast Gradient Method
FGSM	Fast Gradient Sign Method
FL	Federated Learning
GAN	Generative Adversarial Network
IDS	Intrusion Detection System
IoT	Internet of Things
JSMA	Jacobian-based Saliency Map Attack
L-BFGS	Limited-memory Broyden Fletcher Goldfarb Shanno
LLM	Large Language Model
MFAA	Multi-Feature Attention Attack
ML	Machine Learning
MLaaS	Machine Learning as a Service
MLP	Multi-Layer Perceptron
NES	Natural Evolution Strategies
NIDS	Network Intrusion Detection
NLP	Natural Language Processing
PE	Portable Executables
PGD	Projected Gradient Descent
SimBA	Simple Black-box Attack
SPSA	Simultaneous Perturbation Stochastic Approximation
SVM	Support Vector Machine

1 Introduction

1.1 Scope and objectives of deliverable

The purpose of D3.2 is to document the results of “*Task 3.3 A Taxonomy of AI Adversarial Attacks*”. The objectives of the deliverable according to the AIAS project’s Grant Agreement are to carry out a survey of adversarial AI (AAI) attacks of the literature studying a variety of domains such as computer vision, NLP, IoT, CLM, and cybersecurity. The surveyed AAI attacks will be exploited to perform a large-scale taxonomy which will include a categorization of the attacks based on unique traits, such as the attack method and type, the application environment, and the risk.

1.2 Connection to other deliverables and tasks

The current deliverable is directly connected to the following deliverables:

- D2.2 - Specifications & Business cases: influenced the selection of Adversarial AI attacks application domains that this taxonomy examines (Task 2.2) [D2.2].
- D3.3 - Adversarial AI Engine: will utilize the results of the taxonomy to generate AI attacks scenarios (Task 3.4).
- D4.1 - AI-Based Detection of Adversarial Attacks: attacks identified in this taxonomy can be used in the design and development of a DL model that is based on semi-supervised lifelong learning and will detect known and unknown adversarial AI attacks (Task 4.2).
- D4.2 - Mitigation of Adversarial AI Attacks & XAI: will benefit from this task (3.3) since it will conduct research, categorization, and implementation of mitigation techniques against adversarial AI attacks including but not limited to those identified in this taxonomy (Task 4.3).

1.3 Document Structure

This document is organized as follows:

- Section 2 presents the motivation behind this taxonomy.
- Section 3 defines the threat model that this taxonomy employs.
- Section 4 describes the methodology that was leveraged to collect the literature that was reviewed in order to perform this taxonomy.
- Section 5 contains a comparison of the available taxonomy methods in the reviewed literature and outlines the distinctive features of the AIAS Taxonomy.
- Section 6 classifies the Adversarial AI attacks identified in the literature in different categories.
- Section 7 discusses the impact of adversarial examples on AI models.
- Section 8 presents the impact of adversarial AI attacks to various applications and critical sectors as well as systems.
- Section 9 summarizes the conclusions derived from the current taxonomy.
- Section 10 concludes the current deliverable.

2 Motivation

The rapid advancement of Machine Learning (ML) and Artificial Intelligence (AI) has automated and improved system

functions across critical sectors such as industry, healthcare, transportation, and environmental management. These technologies have also transformed fields like finance, digital signal processing, including computer vision (CV), and key areas of cybersecurity, such as intrusion and malware detection. However, their widespread adoption and crucial role in these domains make ML and AI systems increasingly attractive and frequent targets for adversaries. Therefore, protecting these systems is essential not only for preventing harm and financial loss but also for ensuring the reliability of services that modern society depends on and maintaining public trust in increasingly automated infrastructures.

To protect these systems, the attacks they face, and their respective impact must first be studied. In this context, well-structured taxonomies of attacks are crucial, as they clarify the threat landscape and enable systematic risk assessment needed for effective protection. Numerous adversarial AI attack taxonomies exist, covering domains such as computer vision, deep learning, cybersecurity [NIST][PIS][IBI][LIC], and, in some cases, the Internet of Things (IoT) [WAN] and Code Language Models (CLM) [YAN], also referred to as Code LLMs.

Despite these efforts, current taxonomies face a set of challenges which are listed in the following observations (O):

- O1: Taxonomies do not adequately capture the *combined* adversarial landscape involving both IoT systems and (Code Language Models) CLMs, alongside other emerging domains.
- O2: IoT networks now underpin vital services in healthcare, agriculture, industrial automation, and smart cities [XWA] [DTH] [JAK]. Attacks targeting IoT-based AI can therefore lead to devastating real-world consequences, including risks to human safety.
- O3: CLMs have become increasingly central to software development pipelines in academia and industry [JIA]. Adversarial attacks targeting these models carry substantial risks, ranging from code injection and vulnerability propagation to large-scale supply-chain compromise.
- O4: Observations 1-3 highlight the need for a unified taxonomy that simultaneously addresses these rapidly evolving domains while situating them within the broader ecosystem of adversarial AI threats.
- O5: Beyond domain coverage, an effective taxonomy should also reflect how adversaries can interact with AI systems based on their knowledge.
- O6: The *level of knowledge required to execute an attack* is a natural and intuitive organizing principle, as it conveys both the attacker’s capabilities and the practical difficulty of carrying out different attack classes. This dimension also aligns with widely accepted threat models in adversarial machine learning, such as white-box, gray-box, and black-box settings.

The above observations motivate the AIAS project consortium to propose a comprehensive taxonomy that uses adversarial knowledge requirements as its primary organizing axis while integrating key application domains, including Cybersecurity, (Natural Language Processing) NLP, IoT, Computer Vision, CLMs, and industrial AI systems, into a unified and coherent framework. This approach not only bridges gaps in existing taxonomies but also provides a clearer foundation for evaluating risks and designing defenses.

3 Threat modeling

The threat model facilitates the identification of every possible type of threat against an AI/ML model. This model can be implemented by recognizing the capabilities of the adversary and what threats an AI/ML model may face. The following assumptions are made regarding the adversary.

- **Assumption 1-Adversarial capabilities:** The adversary has unlimited computational resources in terms of both software and hardware. The adversary possesses all the necessary skill set and tools to carry out attacks against AI/ML systems and exploit known vulnerabilities.

- **Assumption 2-Adversarial access to the model:** The adversary has physical or remote access to the AI/ML model and can interact with it during the inference or training phase. Interactions can be direct or indirect with the latter encompassing systems with which the model cooperates.
- **Assumption 3-Adversarial knowledge of the model:** The adversary may possess three types of knowledge regarding the model. 1) *Full or insider-level knowledge of the model*; 2) *Partial knowledge of the model* 3) *No knowledge of the model, and the only knowledge they can infer regarding the model is through the output given to a specific input they provided.*

Artificial Intelligence and Machine Learning models face a series of threats that may compromise their operation and result in them behaving in an unexpected manner and producing undesirable outputs or decisions. Bellow a list of threats (T) categories is presented.

- ***T1 – Insider threats*:** Refer to insiders who may be honest but through misconfiguration or errors of the model may lead it to behave unexpectedly or reach undesirable decisions. Another type of insider threats are malicious insiders motivated by gain or other factors that may tamper with or manipulate the model on purpose and disrupt its operations or force it to reach erroneous decision and behave unexpectedly.

- ***T2 - Spoofing:*** The adversary may attempt to deceive a model to misidentify (or misclassify) data, an object, a behaviour or evade detection. In the context of AI/ML, these attacks may also be performed physically by tampering with an object or its appearance.

- ***T3 - Man-in-the-Middle attack (MitM)*:** The adversary may intervene in the communication between a system and the AI/ML model that monitors it. In this attack, the adversary may make tiny and undetectable modifications in the data exchanged during the communication between the system and the AI/ML model. The consequence of such attack can be catastrophic, especially in critical sectors, since if successful can force the model to make erroneous decisions. An MitM attack may also result to the interception of sensitive data to be processed by the AI/ML model.

- ***T4 – Breach of privacy & Data leakage:*** There are two scenarios in which this threat can be realized. In the first scenario, data from the training datasets are revealed accidentally during inference. Conversely, in the second scenario, adversaries craft queries that aim to deceive the model into revealing sensitive information from a model. The second scenario is also known as model inversion.

- ***T5- Compromise of data Integrity:*** The adversary may inject malicious carefully crafted poisonous examples into

the training datasets thus causing the model to make erroneous decisions on specific inputs.

- ***T6-Denial of Service:*** The adversary may perform a SYN flood attack [CMU] against the server on which the AI/ML model is deployed. This attack may prevent the model to receive data that will use to make inferences. Since most AI/ML models require a considerable amount of data to make reliable inferences and predictions this attack can have a severe impact on the model. Consequently, the model will no longer be in position to produce, at least reliable, predictions and estimation.

- ***T7-Supply Chain attacks:*** The adversary may exploit vulnerabilities in plugins or development libraries, used in the model, originating from third parties and obtain a greater degree of access to the model or knowledge.

- ***T8-Model theft:*** An adversary may repeatedly query an AI model to receive outputs and based on them to determine and replicate the model’s functionality. The adversary effectively steals the model without access the source code or training data.

4 Survey-Literature collection methodology

The purpose of this section is to describe the methodology that was utilized for the survey and collection process of the literature. Specifically, it indicates the keywords that were used as search terms along with the online repositories that were surveyed. Moreover, it describes the queries that were deployed along with the selection criteria. The selection criteria include subject of the paper/article, its alignment with survey’s proposed topic, its year of publication and application domains. Additionally, it presents the number of selected articles and the number of survey articles.

- The sites that were used to collect the literature used in this survey are: Google Scholar[1], IEEE explore[2], Elsevier[3], Arxiv[4], SpringerLink[5], ACM Digital Library[6], and Science Direct[7].
- The following queries were used in the above websites to search for related literature: AAI attacks, Adversarial attacks in Machine Learning, Adversarial Machine Learning attacks in cybersecurity, White-box, Grey-box and Black-box attacks in computer Vision, Natural Language Processing, Cybersecurity and IoT.

Based on the above criteria, 46 articles were selected from which 11 are surveys, and 35 of them propose a novel attack. The selected papers are in the research areas of Cybersecurity (8 attacks), Natural Language Processing (3

[1] https://scholar.google.com/

[2] https://ieeexplore.ieee.org/Xplore/home.jsp

[3] https://www.elsevier.com/

[4] https://arxiv.org/

[5] https://link.springer.com/

[6] https://dl.acm.org/

[7] https://www.sciencedirect.com/

attacks), IoT (1 attack), CLM (4 attacks) and Computer Vision (1 attacks) and were published between 2008 and 2025.

5 Comparative Analysis of Existing Taxonomies

5.1 Available taxonomy methods in the literature

Here, we present the most representative adversarial Artificial Intelligence & Machine Learning attack taxonomy methods that are available in the literature.

Yang *et al.* propose in [YAN] categorizing attacks against Code Language Models generated code into three types according to the Confidentiality Integrity Availability (CIA) triad: a) poisoning attacks, which lead to infringement of a model’s integrity and availability, b) evasion attacks that aim to infringe integrity, and c) privacy attacks that target confidentiality.

Another taxonomy approach that focuses on ML-specific and deliberate threats is presented in [KAW]. Specifically, this taxonomy encompasses ML-related threats on three situations in the model’s lifecycle: a) threats to ML-based systems during development, b) threats do ML-based systems during operation, and c) threats to pre-trained models provided for model users. Situations (a) and (b) deal with threats before and during system operation respectively, while (c) deals with pre-trained models alone and not with an ML-based system. Threats during the system’s development are poisoning attacks against the assets used in the development. Such attacks aim to poison the model or the data. During system operation attacks can occur in the form of malicious inputs to the system during operation and include but are not limited to model extraction attacks, evasion attacks, and sponge attacks. Lastly threats to a pre-trained model concern a situation where the attack against the pre-trained model is performed by either the model provider or the user.

Pispa *et al.* proposed a taxonomy methodology in [PIS] which comprises three stages. The first stage indicates which of the three, according to the authors, phases of the AI system the attack or threat affects. The first phase is the development of the AI model, the second is the training and the third is the deployment phase of the AI system. The third phase includes the deployment to production use and the eventual maintenance of the model along with any modifications. The second stage of the taxonomy focuses on the attributes of the AI that a vulnerability undermines. At this stage, the authors propose using a framework of AI trustworthiness that is closely aligned with the ENISA framework to enable closer cooperation between pertinent parties in the AI security landscape. The third stage of the taxonomy assess the degree of degradation of the AI model’s functionalities with respect to the vulnerability’s original creator’s intention. This metric enables the generation of more detailed descriptions of a how a vulnerability affects a model. Bountakas *et al.* performed in [BOU] a domain agnostic taxonomy of attacks against AI/ML models in the domains of audio, cybersecurity, NLP and CV. This taxonomy also encompassed defense strategies from these attacks.

The taxonomy approach by Chaganti [CHA2] performs a classification of attacks using the following criteria: a) aims and objective of the attacker, b) knowledge level of the attacker, c) degree of specilisation regarding the attack target which indicates if the attack targets a specific class or aims to cause general misclassification, d) frequency of the attack which indicates if it occurs one-time or multiple. This taxonomy is oriented for attacks occurring against AI systems deployed in the cybersecurity application domain. Pitropakis *et al.* performed a taxonomy of attacks in ML in CV, cybersecurity and standalone ML models in [PIT]. Other taxonomy surveys were performed exclusively in the

application domains of cybersecurity [PIS, IBI], Deep Learning (DL) [CHA], and CV [LIC, DAI].

Shayea *et al.* proposed in [SHA] a taxonomy method that classifies attacks based on timing, perturbation type, knowledge and goals. This attack taxonomy mentions attacks from the domains of CV, NLP, malware detection, and IoT. Wang et al. [WAN] proposed a taxonomy on adversarial attacks against Cyber Physical Systems, which organized attacks based on the attacker’s knowledge, attacker’s strategy, attacker’s source, attacker’s proximity, attacker’s perceptibility and attacker’s goal. Although this taxonomy is exhaustive in terms of attakcs carried in the application domain of Cyber Physical Systems it does not examine attacks to other application domains.

NIST presented their own taxonomy in [NIST] which seperates attacks into those that target predictive AI systems and those related to generative AI. This taxonomy considers components of AI systems such as data, the model itself, the training process, testing and deploying the model. Additionally, it considers the application contexts into which models can be utilized, such as scenarios involving the deployment of Generative Artificial Intelligence with access to private data or equipped with tools to take actions with real-world consequences.

In short, this taxonomy classifies attacks based on a) the AI system type, b) the stage of the model’s life cycle during which the attack is conducted, c) the attacker’s goals and objectives in terms of the system properties they target, d) the attacker’s capabilities as well as access, e) and the atttacker’s knowledge. This taxonomy does not consider adversarial attacks against IoT ecosystems and CLMs. Additionally, even though this taxonomy is comprehensive and suitable for deployed generative systems it does not categorize attacks using their required level of knowledge as a central organizing axis. Having an understanding of the required level of knowledge is important since it immediately and intuitively indicates how difficult it is to carry out this attack as well as the attack’s severity.

In addition to the above taxonomies, several foundational or frequently referenced classification schemes have shaped the adversarial ML landscape. Papernot *et al.* [PAP3] provided a widely adopted systematization of attacks based on adversary knowledge, goals, and the stage of the ML pipeline targeted. Earlier works such as Barreno *et al.* [BAR] and Huang *et al.* [HUA2] laid the groundwork by introducing the influence/knowledge/goal taxonomy that still underpins many contemporary approaches. Biggio and Roli [BIG3] added a decade-long retrospective that systematizes attack strategies and defenses in real-world settings, especially in cybersecurity and vision domains. These classical taxonomies focus more on high-level security properties rather than application-specific threats or emerging domains like Large Language Model (LLM) generated code, which are addressed in newer taxonomies such as the one introduced in the AIAS project.

5.2 Distinctive Features of the AIAS Taxonomy

The taxonomy proposed in the context of the AIAS project distinguishes itself from all existing taxonomies in several key dimensions. First, in contrast to prior works that are often restricted to specific application domains, such as CV [LIC], network security [IBI], LLMs [CRO], or cybersecurity broadly [PIS, CHA2], the AIAS taxonomy offers a unified classification across a diverse set of emerging and underexplored domains, including cybersecurity, IoT, NLP, CV, CLMs, and industrial systems. To our knowledge, no other existing taxonomy, aside from the relatively limited scope of [SHA], addresses this specific combination of domains. Secondly, the AIAS taxonomy provides fine-grained classification across multiple attack types, covering perturbation attacks, poisoning (including prompt-instruction poisoning), evasion (including physical-world attacks), and domain-specific exploit strategies. This granularity is missing in broader or generic surveys such as [CHA], [PIT], or [PAP3], which do not delve into the nuances of domain-specific behaviours

or modern attack modalities such as CLM-targeted threats. Thirdly, the AIAS taxonomy is explicitly phase-aware, capturing attacks occurring at training, inference, and preprocessing stages, thus reflecting the lifecycle of real-world AI systems. While other works like [KAW] and [PIS] address lifecycle phases, they often do so from a high-level system-development perspective or lack coverage of preprocessing and fine-tuning vulnerabilities, especially in CLMs. Additionally, the AIAS taxonomy is one of the few frameworks that integrates benchmark attacks such as Jacobian-based Saliency Map Attack (JSMA), DeepFool, Basic Iterative Method (BIM), and Mask Domain Generation Algorithms (MASKDG) which are essential for evaluating adversarial robustness but are overlooked in surveys like [SHA] and [BOU]. Moreover, it is the only taxonomy in the table that incorporates attacks against CLM-generated code, a rapidly growing vector of adversarial interest in LLM-integrated environments, which remains entirely unaddressed in prior taxonomies. Taken together, these characteristics position the AIAS taxonomy as a comprehensive, cross-domain, lifecycle-aware, and attack-type-inclusive framework, capable of supporting both theoretical understanding and practical defence analysis across contemporary AI systems.

Table 3. Comparison of taxonomies

Taxonomy method	**Application Domain(s)**	**Scope and coverage**	**Attack phases covered**	**Attack goals**	**Attacker Knowledge**
AIAS – Taxonomy	Cybersecurity, NLP, IoT, CV, CLM, industrial systems	Perturbation attacks in computer vision, poisoning attacks in CLM, evasion attacks in CLM and NLP, evasion attacks in malware detection, physical evasion attacks in CV, evasion attacks in cybersecurity and domain classifiers.	Training and inference phases	Misclassification, compromise availability, integrity and confidentiality of ML models in cybersecurity, NLP, IoT, CV, CLM, increase error rate.	White-box, grey-box, black-box
[YAN]	CLM-generated code	Data poisoning, model poisoning, prompt-instruction poisoning, model poisoning at the fine-tuning stage, model poisoning at the pre-training stages.	Training, inference, preprocessing phases	Compromise the availability/usability of CLMs for all users or a group of users. Compromise the CLMs integrity.	White-box, grey-box, Black-box
[KAW]	CV , cybersecurity, NLP, IoT	Data poisoning attack, model poisoning attack, exploitation of a poisoned model, model extraction attack, evasion attack, sponge attack, information leakage attack.	Development and Operation (Inference) phase	To cause malfunction of the trained model for: a) specific inputs, b) for inputs that contain specific information, c) for unspecified inputs. To obtain a trained model with an unintended functionality. To perform DoS against the model. To embed sensitive information to datasets and disclose them during the model's operation. These goals apply for data poisoning and model poisoning attacks.	White-box and black-box. Regarding grey-box attacks, there is only the definition.
[CHA2]	Cybersecurity	This survey provides a measurement framework that classifies adversarial	Training and Inference	Confidence reduction, misclassification, targeted	

		attacks and, also, provides defense strategies tailored for AI-based security systems.		misclassification, source/target misclassification.	
[LIC]	CV	Pixel attacks, sparse attacks, universal adversarial attacks and style transfer attacks in computer vision.	Inference	Non-targeted attacks in computer vision the purpose of which is the adversarial examples to make the prediction result of Deep Neural Networks (DNNs) change. Targeted attacks in computer vision which aim to successfully classify adversarial examples into a predetermined class. Changing pixel points that are critical to the model’s decision-making process.	White-box and Black-box
[BOU]	Audio, cybersecurity, NLP, CV	Focuses solely on the existing defenses against Adversarial Machine Learning (AML)	Testing, training and inference phase	Misclassification, violation of model’s integrity.	White-box, grey-box, black-box
[PIS]	Cybersecurity	3-stage process of vulnerability location, compromised attributes, possible exploitation.	Development and Inference	Degradation of prediction, accuracy, increase in noise of the results, slowness in response times, excessive data/computational resource usage, Lack of AI model availability.	Does not consider attacker knowledge.
[IBI]	Cybersecurity	Network security and introduction of two classification approaches.	Orthogonal (introduced the concept of problem space and feature space dimensional classification of adversarial attacks in network security.)	Poisoning/Evasion/Oracle.	White-box, grey-box, black-box

[CRO]	LLMs	This survey focuses on threat models where a threat actor leverages LLM-generated text as part of an attack. This involves scenarios in which the attacker attempts to pass off machine text as human and where the detection of LLM-generated text may be used for defensive purposes. The authors do not discuss attacks against natural language generation(NLG) models themselves unless they leverage LLM as part of the attack. This includes using an LLM to produce data for poisoning a model’s training dataset.	Training and Inference	Using LLMs to perform phishing and scamming. Use LLMs to send malicious messages through compromised social media accounts to propagate an exploit to other users (social worms). Perform model poisoning with LLMs. Use LLMs to perform poisoning attacks against machine learning models.	Does not consider attacker knowledge.
[CHA]	Deep Learning	No restriction to specific applications and, also, in a more elaborate manner with practical examples. Too generic.	Training and inference.	Misclassification and confidence reduction.	White-box and Black-box
[DAI]	Vision Language Models (A field that combines NLP with CV)	Focuses on the interplay between visual and textual examples that can negatively impact NLP models.	Training and Inference	Jailbreak, camouflage, and exploitation attacks that aim to generate “not safe for work” images and text, perpetuate harmful stereotypes and to extract personal information without authorization, increase a model’s operational cost.	White-box, grey-box, black-box

[PIT]	Machine Learning	58% of the attacks are oriented against cybersecurity applications (spam filters, Intrusion Detection System (IDS), malware detection) and CV.	Training and inference.	Targeted attacks against a particular sample or small set of samples. Non-targeted attacks against a general category of samples. Misclassification, clustering accuracy reduction, increase both false positives and false negatives rate, and Denial of Service (DoS) in ML.	White-box, grey-box, black-box
[PAP3]	Generic (cybersecurity, CV, NLP, etc.)	CIA-based goals, influence (training/inference), capabilities (white/black-box).	Training and Inference.	Misclassification, confidence reduction, privacy breaches.	White-box, Black-box
[BIG3]	Generic (focus on vision and cybersecurity)	Retrospective view of real-world threats and defensive evolution.	All phases	Targeted/untargeted, DoS, info leakage, data poisoning.	All levels
[BAR]	ML (general)	Exploratory vs. causative, targeted vs. indiscriminate.	Training and inference.	Model failure, integrity breach, data compromise.	Perfect knowledge, limited knowledge
[HUA2]	General ML systems	Early attack taxonomy: adversary's goal, knowledge, and influence.	Training and inference.	Misclassification, confidence degradation.	White-box, grey-box, black-box
[NIST]	Generic (Image, Text, Audio, Video, Cybersecurity)	Data poisoning attack, model poisoning attack, exploitation of a poisoned model, model extraction attack, evasion attack, sponge attack, information leakage attack.	All phases	Misclassification, integrity breach, data compromises confidence degradation.	White-box, grey-box, black-box

6 AIAS Taxonomy Dimensions and Categories

6.1 Categorization based on attack timing

Adversarial attacks in ML can be classified along multiple dimensions, depending on the attacker's goals, capabilities, and interaction with the target system. One of the most common taxonomies distinguishes between white-box and black-box attacks, based on the level of access and knowledge the adversary has about the model's architecture and parameters. However, an orthogonal and equally important classification is based on the phase of the ML lifecycle that the attack targets: "training-time versus inference-time attacks", as depicted in Figure 1.

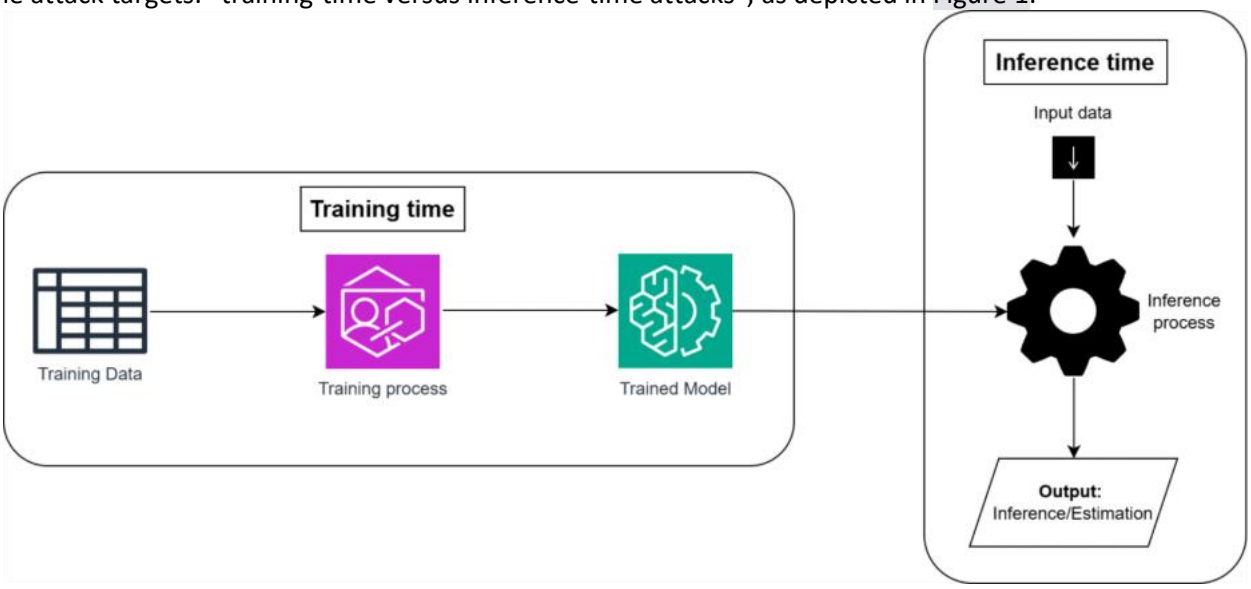

Figure 1: Training vs. inference time attacks

The training phase encompasses all stages involved in building the model, including data collection, preprocessing, learning from data, validation, and final deployment. A training-time attack, therefore, refers to any adversarial manipulation that occurs while the model is being developed or trained, before it is made available for use. These attacks seek to influence the learning process itself, often by corrupting training data or injecting hidden behavior into the model that persists after deployment.

In contrast, the inference phase begins once the model is deployed and begins serving predictions. Inference-time attacks exploit the model without modifying its internal state, instead crafting inputs that lead to incorrect or malicious outputs. These attacks are especially relevant in scenarios where the model is exposed via public application programming interfaces (APIs) or query interfaces, and the adversary has no access to the underlying training process or model internals.

While these two phases provide a useful conceptual boundary, it is important to recognize that they are not always strictly separated in practice. For instance, some adaptive systems may retrain on incoming data or user feedback, allowing a clever adversary to blur the line between inference and training by submitting strategically designed queries that eventually influence future model updates. Nonetheless, this "training-time vs. inference-time" distinction remains a powerful framework for understanding how adversaries interact with ML systems and where appropriate defences must be deployed.

6.2 Training time attacks

A training-time attack is a category of adversarial attack in which the adversary seeks to compromise a ML model by interfering with its training process. These attacks are executed during the model’s learning phase, with the intent of embedding malicious behavior that persists into deployment [CHV]. The adversary may manipulate the training dataset by injecting poisoned samples, modifying labels, or subtly altering feature values, or directly influence the learning algorithm, for instance by altering the optimization objective or loss function. The main objective is to cause the model to learn incorrect or adversary-controlled decision boundaries, which may result in performance degradation, systemic bias, or covert vulnerabilities such as backdoors. Because the malicious behavior is introduced before the model is ever deployed, training time attacks are particularly insidious: they can evade post-training evaluation metrics and remain dormant until activated under specific conditions. This makes them a serious threat in environments where training data is collected from untrusted or distributed sources, such as Federated Learning (FL), crowdsourced datasets, or public data repositories [GOL].

6.3 Inference time attacks

Inference-time or evasion attacks occur after a ML model has been deployed and is actively making predictions. These attacks exploit latent vulnerabilities that have been embedded or are already inadvertently present [SZE] into the model, either during training or post-deployment, allowing an adversary to selectively manipulate the model’s behavior at test time without needing to alter the model’s architecture or weights in real time. In a typical backdoor scenario, the model behaves normally on most inputs, preserving high accuracy and appearing trustworthy. However, when a specific trigger pattern (e.g., a particular pixel arrangement in an image or a keyword in text) is present in the input, the model is intentionally misled to produce a malicious or incorrect output. These attacks are especially dangerous because they provide covert and targeted control over the model’s predictions using only inference-time access, making them difficult to detect with standard testing procedures. Moreover, since the adversary can activate the trojan behavior on demand, such attacks can be used for impersonation, misclassification, or access control evasion in systems like facial recognition, autonomous driving, or spam filtering. The effectiveness and stealth of inference time attacks underscore the necessity for robust defenses like input sanitization, anomaly detection, and backdoor detection techniques during model validation and deployment.

Consider a typical deployment scenario in which a machine learning model is exposed via an API endpoint, common in cloud-based services, commercial ML platforms, or Machine Learning as a Service (MLaaS) environment. In this black-box setting, the adversary does not have access to the internal architecture, parameters, or gradients of the model. However, they can repeatedly query the model with inputs of their choosing and observe the corresponding outputs, which may include predicted class labels, confidence scores, or other metadata. This seemingly limited access can still be exploited for powerful inference-time attacks. For example, the attacker may craft a sequence of strategically varied inputs and analyze the output patterns to infer sensitive information about the model or its training data. In model extraction attacks, this process can be used to reconstruct an approximate replica of the target model. In membership inference attacks, the adversary attempts to determine whether specific data points were part of the model’s training set. Even without visibility into how inputs are processed internally, the attacker leverages the statistical behavior of the output space to deduce valuable internal characteristics. This highlights a critical vulnerability of deployed ML systems: their predictive outputs alone may leak private, proprietary, or sensitive information, especially in scenarios involving high-stakes data like medical records, facial biometrics, or financial transactions.

6.4 Categorization based on attacker knowledge

The knowledge an adversary has for a system plays a crucial role in designing, implementing and carrying out an attack. The attacks oriented against AI models are no exception and based on the amount of knowledge adversary has, AI attacks can be classified into three categories: a) white-box attacks, b) black-box attacks, and c) grey-box attacks [BOU].

6.5 White box attacks

White-box attacks refer to the case in which an adversary has complete knowledge of the system, the AI algorithm’s architecture, the hyper-parameters and the utilized training data. Although, the availability of knowledge the adversary needs to have to carry out these attacks is rare in real life it does not mean that they are impossible to occur. The adversary can obtain this knowledge either via reverse engineering the model or by a malicious or honest insider. Through reverse engineering, adversaries can recover a set of model’s components such as architecture or hyperparameters and gain white-box access. Considering the above, one can infer that white-box attacks are the most potent type of AI attacks and are usually deployed to evaluate the robustness of AI attack models and defenses.

Several white-box attacks have been discovered and document in the literature that can affect applications such as CV [LIC], IoT networks, and NLP. One attack that may affect CV is the *Fast Gradient Sign Method (FGSM)* which exploits the nonlinearity of Deep Neural Networks (DNNs) and by utilizing a small perturbation to the input data, it is enough to mislead the classification results of DNN. FGSM [GOO] is a well-known white-box attack technique that exploits the nonlinearity of DNNs. It demonstrates that even a small perturbation added to an input can significantly alter the classification results of a DNN. The attack is formulated as:

$$x_{\{adv\}} = x + \epsilon * sign(\nabla_x J(\theta, x, y))$$

Equation 1 FGSM formula

Where ϵ represents the perturbation magnitude, *sign()* denotes the sign function, *J* is the cross-entropy loss function, and θ refers to the DNN's weights. The variable *y* represents the true label of the input image *x*. FGSM operates by first computing the gradient of the input image concerning the loss function. The sign function is then applied to determine the attack direction. Using gradient ascent, a perturbation is introduced to the image, increasing the discrepancy between the predicted labels and the true labels. Notably, FGSM applies a uniform perturbation size to each pixel and requires only a single iteration to complete the attack, making it an efficient white-box attack method [GOO]. The FGSM method and its derivatives are inference time attacks.

An example of applying the FGSM concept on NLP is TextFool [BIN], which approximates the contribution of text items that possess significant contribution to the classification of text. Instead of using the sign of cost gradient in FGSM, this attack considers the magnitude. The authors presented three attacks, which are insertion, modification and removal. Another attack similar to TextFool that is oriented to the text domain is presented in [SUR] by Suranjana *et al.,* which provides an attack that follows the removal-addition-replacement pattern. An attack that follows that pattern first tries to remove an adverb (denoted *wi*) that contributed the most to the text classification task. In case the output sentences in this step have incorrect grammar, the method will insert a word *pi* before *wi.* The word *pi* is chosen from a candidate pool, which contains synonyms misspelled words, typos and genre specific keywords. If the output cannot

satisfy the highest cost gradient for all the pi then the attack replaces *wi* with *pi*. Adverbs are prioritized because they function as high-impact modifiers of intensity or sentiment (e.g., replacement of “extremely fair” with “fair” shifts ambiguous to negative) yet are often grammatically optional, allowing deletion without syntactic disruption.

On the malware detection front, FGSM has been used by the authors of [ABD] to generate binary-encoded adversarial examples. In malware detection, there are Portable Executables (PE), which are represented by binary vectors. In these vectors 0 and 1 indicate whether the PE is absent or present. By using the PE vectors as features, DNNs that detect malware can identify the malicious software. The authors of this work incorporated four bounding methods to generate perturbations in order to preserve the functionality of the adversarial examples. The first two methods utilize FGSMk which is a multi-step variant of FGSM. To restrict the perturbations in a binary domain the authors employ deterministic rounding dFGSMk and randomized rounding rFGSMk. Another example of attacking DNNs that detect malware is presented by Ishai *et al.* [ISH]. In this work the attacker makes perturbations on the embedding presentation of the binary sequences and reconstructs the perturbed example to its binary representation. Specifically, the attacker appends a uniformly random sequence of bytes to the original binary sequence. Then a new binary is added to the embedding and performs FGSM only on the payload’s embedding. The perturbation is performed iteratively until the detector outputs an incorrect prediction. Since the perturbation is only performed on the payload, instead of the input, this method will preserve the functionality of the malware.

The *BIM attack* builds upon FGSM by addressing the model's nonlinear nature, where gradients can fluctuate significantly within a small range. Since FGSM applies a single-step perturbation, the magnitude may sometimes be too large, causing adversarial examples to fail in successfully misleading the model. BIM refines this approach by applying multiple small perturbation steps iteratively to enhance attack effectiveness. The iterative update rule for generating adversarial examples is given by:

$$x_{adv}^{t+1} = x_{adv}^{t} + a * sign\left(\nabla_{x_{adv}^{t}}(\theta, x_{adv}^{t}, y)\right)$$

Equation 2 BIM formula

Where a is the size of the perturbation added in each iteration. Experimental results have indicated that BIM is more competitive compared FGSM [KUR]. BIM is an inference time attack.

Another white-box attack is *DeepFool* which was introduced by Moosavi *et al.* [MOO] and is designed to generate adversarial examples with minimal perturbation. This method estimates the shortest distance from an input sample to the nearest decision boundary and utilizes this distance as a measure of the model's robustness. Specifically, DeepFool iteratively applies small perturbations to the original image, gradually pushing it toward the decision boundary. Once the boundary is crossed, the accumulated perturbations are combined and applied to the original image, resulting in an adversarial example. The DeepFool attack is performed during the model’s inference time.

The *Projected Gradient Descent (PGD) attack*, introduced by [MAD], is an extension of BIM with key modifications. Unlike BIM, PGD increases the number of iterations and begins by initializing a random perturbation added to the original image. The perturbation magnitude applied at each step is a user-defined hyperparameter, independent of

the total number of iterations. After each update, the perturbations are projected back within a predefined threshold to ensure they remain within the allowed perturbation range. PGD is widely regarded as one of the most powerful white-box attack methods and is frequently used to evaluate model robustness. The attacker carries out the PGD attack during inference time.

Carlini and Wagner proposed an attack in [CAR] which is optimization-based and uses the following optimization objective function:

$$\min ||\delta||_p + r \times F(x + \delta), s.t. x + \delta \in [0,1]^m$$

Equation 3 Carlini and Wagner formula

In this function *r* is a hyperparameter, *m* denotes the image channel dimension, and δ represents the intensity of perturbation. *F(x+δ)* has the following definition:

$$F(x + \delta) = \max(\max\{Z(x + \delta)_i : i \neq t\} - Z(x + \delta), -k)$$

In this equation *Z(x)* signifies the probability output of the classifier and the hyperparameter *k* constrains the confidence that the adversarial examples are misclassified as label *t*. The application of this attack has demonstrated that the adversarial examples it generates are able to achieve high attack success rate even if defensive distillation methods. However, the optimization process is time-consuming because it involves finding suitable hyperparameters. This attack is conducted during the inference phase of the model.

Hu *et al.* present in [YUC] a white-box attack, that is conducted during the inference phase, using a naturalistic adversarial patch. This approach, unlike previous patch-based attacks, creates visually inconspicuous patches that can deceive DL-based object detectors in real-world settings. An interesting aspect of this attack is that it is physical since the adversarial example is printed and added to the target object as a sticker in some cases. To conduct this attack the following methodology is used. First a pretrained Generative Adversarial Network (GAN) is used to generate an image intended to serve as the adversarial patch. These images are designed to appear natural or realistic, unlike random noise or abstract patterns used in other adversarial methods.

Suciu *et al.* proposed an inference time white-box attack in [SUC] oriented towards malware AI detection models shifting away from the trend of employing white-box attacks against computer vision. Specifically, in this paper, the authors explored evasion attacks by focusing on the MalConv byte-based convolutional neural network for malware detection. MalConv reads up to 2MB of raw byte values from a PE file as input while for smaller files it appends a distinguished padding token and truncates extra bytes from larger files. The fixed-length sequences are then transformed into an embedding representation and these embeddings are then passed through a gated convolutional layer. Afterwards, they are passed through a temporal max pooling layer and then are classified through a final fully connected layer. The authors use *Append and Slack attacks* to attack MalConv. Append-based attacks focus on the semantic integrity constraints of PE files and append adversarial noise to the original file.

There are four principal types of append attacks. The first is the *Random Append* attack, where byte values sampled from a uniform distribution are appended to the end of a PE file. The second, *Gradient Append*, leverages the input

gradient to guide the modification of appended byte values, making the attack more targeted. A third variant, known as *Benign Append*, demonstrates MalConv’s susceptibility, particularly that of its temporal max-pooling layer, to adversarial manipulation that reuses benign byte sequences appended at the end of a file. Finally, the *FGM Append* attack is introduced as a “one-shot’’ alternative motivated by the observation that Gradient Append converges more slowly as the number of appended bytes increases. This method adapts the fast gradient method to generate the appended bytes in a single optimization step.

Regarding *Slack attacks* the authors deploy Slack FGM [SUC] which defines a set of slack bytes where an attack algorithm is allowed to freely modify bytes in the existing binary file without breaking the PE. Although this is a white-box attack, its adversarial example generation methods can be transferred to black-box attacks.

The *Limited-memory Broyden Fletcher Goldfarb Shanno (L-BFGS) [SZE] attack* is an inference phase white-box attack that adds perturbation to images to deceive DL models and lead to misclassifications. The attacker generates adversarial examples by adding perturbations to an image with the purpose of reducing the added perturbation r under $L_2 = \min||r||_2$, $f(x+r) = y$, where r is the perturbation, f is the loss function of the DL model, x is the original image, and y is the incorrect prediction label. A disadvantage of the L-BFGS method is that even though it is efficient at generating adversarial examples, it is computationally ineffective.

In the application domain of CLM there is a variety of white-box attacks, which includes poisoning and evasion attacks. In CLM model, *poisoning attacks* the adversary has control over the training/fine tuning process of the victim CLM in way that the latter’s availability will be degraded for downstream users. One such attack is proposed by Schuster *et al.* in [SCH] that investigated model poisoning availability attacks to make the victim CLM output unsafe code for downstream users. This attack was validated by fine-tuning the victim CLM on poisoned data with a few epochs, which resulted in the victim CLM generating unsafe code with a rate reaching 100% on both Pythia [BID] and GPT-2 code generation models. On the evasion attacks front, there are the *Embedding Level Gradient-Based Approach (ELGB) attack* [HZH] and the *continuation-based approach attack* [SRI2]. In the ELGB attack the adversary computes the model gradient and afterwards projects the gradient back to the input token level with the nearest search. In the continuation-based attack the adversary formalizes the discrete source code adversarial examples into continuous ones and applies the gradient-based solver to acquire optimal solutions. In the application domain of malware detection systems, MalPatch [ZHA] is a white-box inference time adversarial attack that used against DDN-based malware detection tools. This attack produces attack-independent adversarial patches to attack various types of detectors and works by injecting the patches into malware samples.

Papadopoulos *et al.* in [PAP2] presented an attack against Network Intrusion Detection System (NIDS) in IoT networks which includes two main approaches. The first approach focuses on label poisoning with the purpose of causing incorrect classification by the model while the second approach used FGSM to evade detection measures. Specifically, this work evaluated the robustness of the Bot-IoT dataset against label noise attacks using a Support Vector Machine (SVM) model while FGSM was used to generate adversarial examples against binary and multi-class Artificial Neural Networks (ANNs).

The authors of [OUA] presented a white-box evasion attack oriented towards malware detection RNN, DNN and CNN models. To attack the models the authors deployed FGSM, PGD and the BIM attacks. The evaluation of the attacks

showed that the PGD attack is the most potent compared to the rest as results showcase that it brings the greatest reduction in the accuracy of every model.

6.6 Black box attacks

Black-box attacks operate without any internal access to the target model, relying solely on query-based feedback such as predicted labels or output confidence scores. These attacks are especially relevant in real-world deployment scenarios, including MLaaS, where models are accessible only through remote APIs. In such settings, the adversary must craft adversarial examples without gradients or architectural knowledge, observing only the model's output $f(x+\delta)$ for a perturbed input $x+\delta$. This fundamental constraint renders traditional white-box optimization techniques inapplicable and drives the development of alternative methodologies. Although earlier literature commonly divided black-box attacks into transfer-based and query-based categories, this binary taxonomy has become increasingly limited as the field progresses. Modern approaches often blend these paradigms, using surrogate models to generate initial perturbations, refining them via query-efficient feedback loops, or incorporating structural and frequency-aware priors that transcend classical boundaries. As a result, contemporary research points toward the need for a more expressive and mechanism-driven taxonomy that better captures the evolving landscape of black-box adversarial attacks.

Black-box adversarial attacks can be classified according to the level and type of access an attacker has to the target model's outputs, and a refined taxonomy begins with *Substitute Model-Based (Transfer-Based) Attacks*. In this setting, the adversary has no direct access to the target model's outputs and instead trains a local surrogate model designed to approximate the behavior of the target system. Adversarial examples are then generated using standard white-box techniques such as FGSM or PGD on this surrogate and deployed against the actual black-box model, with their effectiveness relying on the well-established property of adversarial transferability. Various extensions strengthen this approach by incorporating structural priors, such as convolutional feature patterns or feature-space alignment, to improve surrogate fidelity and enhance transfer success [DON, SHI, ZHO, PAP].

One of the seminal contributions to the field of adversarial machine learning was introduced by Papernot *et al.* [PAP], who proposed the first practical black-box attack against remotely hosted DNNs without requiring access to the model architecture, parameters, or gradients. In this groundbreaking work, the authors demonstrated that an adversary, which is limited solely to observing predicted labels for chosen input queries, can still effectively induce misclassifications in a target DNN. The core of their approach lies in constructing a local substitute model that approximates the decision boundaries of the target DNN. This is achieved by generating synthetic inputs through a Jacobian-based data augmentation strategy, querying the target model to obtain their labels, and training the substitute model on this labelled dataset. Once trained, the substitute model is used to craft adversarial examples via white-box techniques, which are then transferred back to the original black-box model. This transferability method proves highly effective. In a real-world, blinded evaluation, the authors attacked a DNN hosted by MetaMind (a deep learning API provider) and observed a misclassification rate of 84.24% on adversarial examples generated via their substitute model. To further demonstrate the generality of their approach, they conducted attacks against models hosted by Amazon and Google in their respective AI cloud-based services, using simple logistic regression substitutes. The results were striking: 96.19% of adversarial examples fooled Amazon's model, and 88.94% succeeded against Google's. Additionally, the study revealed that their black-box strategy could bypass several existing defense

mechanisms, previously thought to strengthen models against adversarial inputs. This work not only highlighted the vulnerability of machine learning systems in practical deployment scenarios but also laid the foundation for an entire line of research into black-box adversarial attacks and the development of more robust defenses.

Authors of [DON] introduced *Multi-Feature Attention Attack (MFAA)*, a novel method designed to enhance the transferability of adversarial examples by strategically manipulating multi-layer feature representations within deep neural networks. The core premise of MFAA is to disrupt category-relevant features while preserving object-specific ones, thereby generating adversarial perturbations that generalize more effectively across models.

MFAA comprises three main components:

1. *Layer-Aggregation Gradient (LAG):* MFAA first computes a set of guidance maps by aggregating gradients across multiple layers of the network. These maps capture the relative importance of features at different semantic levels, enabling a more holistic view of the model's attention distribution.
2. *Ensemble Attention (EA):* using the guidance maps, MFAA constructs an ensemble attention mechanism that emphasizes object-specific features common across models, while suppressing model-specific features that do not contribute to generalization. This step aims to craft perturbations aligned with features that are consistently influential across different architectures.
3. *Iterative Attention Perturbation:* finally, MFAA iteratively perturbs the ensemble attention maps, refining the adversarial examples to maximize their transferability to black-box models. The optimization process ensures that the crafted adversarial perturbations remain effective even under defense mechanisms. Empirical evaluations conducted on the standard ImageNet benchmark demonstrate that MFAA significantly outperforms existing state-of-the-art transferable attacks. Specifically, MFAA increases the average attack success rate from 88.5% to 94.1% in single-model black-box settings, and from 86.6% to 95.1% in ensemble-model black-box scenarios, even when target models employ defense strategies.

In this work [SHI], the authors propose the *Curls & Whey black-box adversarial attack* to address two primary limitations observed in existing iterative attack methods: limited transferability and suboptimal noise efficiency. The proposed method consists of two key stages, (1) Curls Iteration and (2) Whey Optimization, each designed to enhance the effectiveness and robustness of adversarial perturbations.

- *Curls Iteration:* this phase introduces a novel iterative strategy that alternates between gradient ascent and descent to 'curl' the optimization trajectory. By doing so, the method captures a richer set of adversarial directions, thereby improving the diversity and transferability of the generated adversarial examples. Moreover, this iterative mechanism helps mitigate the diminishing marginal effect, a common issue in traditional gradient-based attacks where successive iterations yield progressively smaller improvements.
- *Whey Optimization:* following the generation of initial perturbations, this refinement stage aims to reduce the magnitude of noise, particularly in terms of the ℓ_2 norm, without compromising adversarial effectiveness. The approach leverages the robustness of adversarial examples, identifying and eliminating redundant noise components to produce more compact and efficient perturbations.

Extensive empirical evaluations on the ImageNet and Tiny-ImageNet datasets demonstrate the superior performance

of the Curls & Whey attack. The method achieves a substantial reduction in perturbation norm while maintaining high attack success rates. Additionally, it exhibits strong transferability against both ensemble models and defenses based on adversarial training. The approach is further extended to targeted misclassification settings, effectively lowering the difficulty of conducting targeted attacks in a black-box context.

In [ZHO], Zhong and Deng conduct a comprehensive investigation into the transferability of adversarial attacks in the context of face recognition systems. They first demonstrate that feature-level attack methods consistently outperform label-level approaches in terms of transferability, highlighting the importance of attacking deep representations rather than output labels. To further enhance the transferability of feature-level adversarial examples, the authors introduce dropout face attacking network (DFANet), a novel dropout-based technique applied within convolutional layers. By introducing stochasticity during training, DFANet increases the diversity of surrogate models, effectively simulating the behavior of an ensemble without the computational overhead. This mechanism enables adversarial examples to generalize more effectively across different target models. Extensive experiments conducted on state-of-the-art face recognition systems, spanning a range of training datasets, loss functions, and network architectures, confirm that DFANet significantly improves the transferability of existing attack methods. Furthermore, when applied to the Labeled Faces in the Wild (LFW) dataset, DFANet enables the generation of a new set of adversarial face pairs capable of successfully attacking four commercial face recognition APIs without requiring any query access. To support ongoing research on the robustness and defense of deep face recognition systems, the authors publicly released this adversarial dataset under the name TALFW, providing a valuable benchmark for evaluating the vulnerability of real-world face recognition platforms.

- *Query-Based Attacks:* these methods require direct interaction with the target model, and can be further divided based on the type of output the attacker receives:
    - *Score-Based Attacks:* the attacker has access to real-valued outputs (e.g., softmax probabilities, logits). These scores enable the use of gradient-free optimization methods, such as numerical approximation, Natural Evolution Strategies (NES)[WIE] and Simultaneous Perturbation Stochastic Approximation (SPSA)[SPA] that estimate the direction of adversarial perturbation through randomized input sampling and model queries, enabling attackers to iteratively craft adversarial examples even when direct access to the model’s gradients is unavailable, or Bayesian optimization [ILY, CHE1, CHE2].
    - *Decision-Based Attacks:* the attacker only observes the final predicted class label (hard-label setting). Without gradient information, the attack must rely on model queries and decision boundaries, typically using methods like *Boundary Attack* or *HopSkipJumpAttack* to gradually refine perturbations [AND, NMA]. In this work, Ilyas *et al.* [ILY] introduce a refined characterization of black-box adversarial settings by defining three realistic threat models that more accurately reflect the constraints faced in real-world scenarios: the query-limited setting, the partial-information setting, and the label-only setting. Each model captures a progressively more restrictive environment in which adversaries have limited access to the target classifier's outputs. To address the challenges posed by these constrained settings, the authors develop novel black-box attack strategies that remain effective where traditional methods fail. These attacks are specifically designed to operate under practical limitations, such as restricted query budgets, incomplete output probabilities, or access to only the final classification label. Empirical results demonstrate the success of these methods in attacking ImageNet classifiers

under the proposed threat models. Moreover, the authors also present a targeted black-box attack against the commercial classifier of Google named Cloud Vision API, successfully bypassing the limitations imposed by query constraints and partial or minimal output information. This work marks a significant advancement in adversarial robustness research by bridging the gap between theoretical attack models and real-world deployment scenarios. Chen *et al.* [CHE1] propose HopSkipJumpAttack, a family of efficient decision-based adversarial attack algorithms that operate with minimal model information, relying solely on binary output indicating class membership. The core innovation of this method lies in a novel gradient direction estimation technique that leverages only binary decisions at the model's decision boundary, enabling effective adversarial perturbation generation even in extremely restrictive black-box settings. The proposed attack family supports both untargeted and targeted variants and is specifically optimized for similarity evaluated with respect to both the L_2 and L_∞ norms. A theoretical analysis of the gradient estimation method and its convergence behavior is also provided, offering insights into the efficiency and robustness of the attack. Empirical evaluations show that HopSkipJumpAttack requires significantly fewer model queries compared to prior state-of-the-art decision-based attacks, while maintaining or exceeding their success rates. Moreover, the attack exhibits strong performance even when applied to models fortified with widely used adversarial defenses, highlighting its practical applicability and robustness against defense-aware systems.
In this work, Cheng *et al.* [CHE2] address the challenge of adversarial attacks in the hard-label black-box setting, one of the most restrictive and practical threat models, where the attacker has no access to model gradients or confidence scores, only the final predicted label is observable for each query. Existing algorithms for this setting often require tens of thousands of queries to successfully generate a single adversarial example, making them inefficient for real-world applications. Building on a prior optimization-based framework, the authors propose Sign-OPT, a novel and query-efficient black-box adversarial attack. Instead of estimating full gradient vectors, as done in standard zeroth-order optimization, Sign-OPT innovatively estimates only the sign of the directional derivative in any chosen direction, requiring only a single query per direction. This key insight drastically reduces the query complexity while still enabling effective optimization in the hard-label setting. Theoretical convergence guarantees are provided for the algorithm, and empirical evaluations on standard benchmarks, such as MNIST[8], CIFAR-10[9], and ImageNet[10], demonstrate that Sign-OPT consistently outperforms prior state-of-the-art methods. Specifically, it achieves 5 to 10 times fewer queries on average while also producing adversarial examples with smaller perturbation magnitudes, highlighting both its efficiency and effectiveness in practical attack scenarios. In this work, Ma *et al.* [NMA] propose a novel framework to enhance query efficiency in black-box adversarial attacks by introducing a generalized substitute model, referred to as the Simulator. Traditional black-box attacks often rely on

[8] https://www.tensorflow.org/datasets/catalog/mnist

[9] https://www.cs.toronto.edu/~kriz/cifar.html

[10] https://www.image-net.org/

model stealing, wherein a substitute model is trained to approximate the behavior of a target model through repeated queries. However, this process incurs high query complexity and remains vulnerable to defensive mechanisms such as query monitoring and throttling. To address these limitations, the authors introduce a meta-learning-based approach that trains the Simulator to generalize across a wide variety of unseen target models. The training dataset is constructed by aggregating query sequences and responses generated from attacks on multiple existing networks. The learning process optimizes a mean squared error-based knowledge distillation loss, minimizing the discrepancy between the Simulator’s outputs and those of the sampled target networks. During meta-training, meta-gradients of this loss are accumulated over multiple tasks to iteratively refine the Simulator’s parameters. Once trained, the Simulator can accurately approximate the decision behavior of previously unseen models using only a limited number of queries. Consequently, a substantial portion of future attack queries can be redirected to the Simulator, significantly reducing the number of interactions with the actual target model. Experimental results on CIFAR-10, CIFAR-100, and TinyImageNet, a branch of ImageNet shown above, demonstrate that the proposed method achieves orders of magnitude lower query complexity compared to baseline model stealing techniques, without sacrificing attack effectiveness. This work marks a significant advancement in practical black-box adversarial attack strategies by making them more query-efficient and scalable.

- *Hybrid Attacks:* recent approaches combine the strengths of both substitute-based and query-based methods. Specifically, *Transfer-Augmented Attacks*, which begin with perturbations generated from a surrogate model, and then refine or filter them using limited queries to the target model. This balances query efficiency with attack efficacy [DON, SHI].
    - *Query + Transfer-Based Attack:* use initial guesses from transfer-based methods and optimize further through targeted queries, often leading to more robust attacks with fewer total queries [DIN, CAI]. In this work, *Ding et al.* [DIN] propose a low-query black-box adversarial attack that leverages the principle of transferability by integrating both optimization-based and transfer-based strategies. While prior black-box attacks often suffer from either excessive query complexity, suboptimal success rates, or high distortion, the proposed method addresses these limitations by striking a balance between query efficiency, attack success, and perturbation quality. The core idea is to fully exploit surrogate models, which are used to generate transferable adversarial perturbations with reduced dependency on queries to the actual target model. These initial perturbations are then refined via an optimized objective function, guiding the attack more precisely toward the target model's decision boundary with minimal additional queries. Extensive experiments conducted on benchmark datasets, such as MNIST, CIFAR-10, and ImageNet, demonstrate the effectiveness of the proposed method. The results show that the attack achieves a success rate exceeding 98.5% while requiring less than 5% of the queries used by several state-of-the-art black-box attacks, all the while maintaining low perturbation distortion. This work contributes a practical and highly efficient framework for real-world adversarial attack scenarios, especially where query access to the target model is limited or monitored. In [CAI], Cai *et al.* introduce *Blackbox Attacks via Surrogate Ensemble Search (BASES),* a novel and highly query-efficient framework for generating black-box adversarial examples. Their approach addresses the limitations of existing attack paradigms: while transfer-based attacks require

no access to the target model but typically yield lower success rates, and query-based attacks achieve higher success but at the cost of numerous queries, BASES aims to combine the strengths of both. At the core of BASES is a perturbation machine, which generates adversarial examples by optimizing a weighted loss function over a fixed ensemble of surrogate models. Crucially, the method searches over the weight space, which has low dimensionality, corresponding to the number of surrogate models, thus requiring very few queries to tune the loss function toward the behavior of the target (victim) model. This low-dimensional optimization enables BASES to achieve targeted black-box attacks with as few as 3 queries per image, and untargeted attacks with just 1–2 queries per image, significantly outperforming prior methods in terms of query efficiency. Experiments on ImageNet classifiers, including VGG-19[11], DenseNet-121[12], and ResNeXt-50[13], validate the high effectiveness of the method. Notably, BASES also achieves strong attack performance against the Google Cloud Vision API, reaching a high untargeted attack success rate with an average of only 2.9 queries per image. Moreover, the perturbations generated by BASES exhibit strong transferability and can be used effectively in hard-label black-box scenarios. The method also generalizes well beyond classification, demonstrating successful attacks on object detection models, highlighting its versatility across a range of machine learning tasks. Despite the diversity of methods, black-box attacks share several common limitations. Compared to their white-box counterparts, they generally exhibit lower *Attack Success Rates* (ASR) [AND, MOO], reduced adaptability even in simplified settings [WUC], and produce more perceptible perturbations [CAI]. Generator-based techniques, while promising, often involve trade-offs between computational overhead and transferability, limiting their practicality in real-time or resource-constrained scenarios [FEN, YIN].

In [FEN] Feng *et al.* propose a novel method for enhancing the effectiveness of black-box adversarial attacks by introducing a robust mechanism *for Conditional Adversarial Distribution (CAD) transfer* that mitigates the limitations caused by surrogate biases. In typical black-box settings, the attacker only has access to the predicted outputs (e.g., class labels) of the target model, with no access to internal parameters or training data. A common strategy to boost attack performance is to leverage adversarial transferability from surrogate models. However, discrepancies in architecture or training data between surrogate and target models can significantly degrade this transferability. To address this challenge, the authors introduce a method that partially transfers the parameters of the CAD learned from white-box surrogate models, while adapting the remaining parameters through query-based feedback from the black-box target model. This hybrid mechanism allows the attack to retain the beneficial components of transferability while dynamically adapting to the specific characteristics of the target model, thus reducing the negative effects of surrogate biases. The approach maintains a flexible and adaptive CAD that can be conditioned on any benign input sample, improving the attack's

11https://docs.pytorch.org/vision/main/models/vgg.html

[12] https://keras.io/api/applications/densenet/

[13] https://docs.pytorch.org/vision/main/models/resnext.html

ability to generalize and succeed across diverse target models. Extensive experiments conducted on standard image classification benchmarks and real-world APIs demonstrate that the proposed method achieves superior black-box attack performance, both in terms of success rate and query efficiency, compared to existing baselines. Yin *et al.* propose in [YIN] a generalizable black-box adversarial attack framework that significantly reduces query complexity by leveraging meta-learning and two forms of adversarial transferability: example-level and model-level. In black-box settings, where model parameters are inaccessible and only limited feedback is available via queries, conventional query-based attacks often require a large number of queries to generate successful perturbations for each input, limiting their practicality. To overcome this challenge, the authors treat the adversarial attack on each benign input as an individual meta-task and train a meta-generator that learns to produce effective perturbations conditioned on benign inputs. During deployment, this generator can be rapidly fine-tuned using a small number of queries on the new target example, while also drawing from feedback obtained in historical attack tasks, thereby exploiting example-level adversarial transferability. Given that meta-training on the target model would be query-intensive, the authors further reduce the cost by employing model-level transferability: they train the meta-generator on a white-box surrogate model, which then serves as an initialization for attacking the black-box target model. This hybrid strategy enables the attack to generalize across both unseen examples and unknown target models. The proposed framework is modular and attack-agnostic, allowing it to be integrated with existing query-based attack methods to enhance their performance. Experimental evaluations across a variety of datasets and models demonstrate that the method substantially lowers query requirements while maintaining or improving attack success rates, establishing its effectiveness for practical black-box adversarial scenarios.

The traditional access-based taxonomy explains *what* information an attacker can obtain but not *when* or *how* the attacker uses computational resources. To bridge this gap, black-box attacks can also be classified based on the timing of their computational effort, i.e., whether most of the optimization occurs offline (pre-query, surrogate or model training) or online (query-driven inference-time adaptation). This timing-based view complements rather than replaces the feedback-based taxonomy.

Substitute model-based attacks predominantly belong to the *offline* category. These techniques require training a surrogate model in advance to approximate the decision behavior of the target, after which adversarial examples are generated locally using white-box methods. Since adversarial inputs are precomputed, the inference-time interaction with the target model is minimal or even unnecessary.

Similarly, transfer-augmented and hybrid approaches also involve a substantial offline phase, where perturbations or latent representations are initialized using one or more surrogate models. However, unlike pure transfer attacks, these methods typically incorporate limited inference-time querying to refine or adapt perturbations to the target model, thereby combining offline preparation with online adaptation.

In contrast, query-based attacks, including both score-based and decision-based variants, are fundamentally *online in nature*. They craft adversarial examples *during inference* by interactively querying the target model to estimate gradient directions, approximate decision boundaries, or refine adversarial perturbations. Because

they depend on real-time model feedback, these attacks incur higher query complexity, increased detectability, and stricter operational constraints.
This timing-based categorization highlights key trade-offs. Offline-heavy attacks tend to be stealthier and less query-intensive but rely on transferability and accurate surrogate modeling. Online-focused attacks, while flexible and architecture-agnostic, often face constraints related to query limits, latency, and defensive monitoring. Hybrid strategies balance these dimensions by integrating surrogate knowledge with selective querying during inference.
Rather than re-examining each attack method in detail, we now reinterpret our previously established taxonomy, substitute-based, score-based, and hybrid attacks, through this computational timing framework. This mapping clarifies how different techniques distribute their effort across training and inference phases and reveals practical implications for adversarial robustness and real-world deployment.

Substitute model-based and transfer-based attacks naturally fall under the offline or training-time category, as they rely on constructing a surrogate model trained to approximate the target decision boundary using initial queries or public data [PAP]. Once trained, adversarial examples are crafted using white-box methods (FGSM, PGD, CW) and subsequently transferred to the target model, often without requiring any additional queries at inference time. The computational effort, therefore, is concentrated before deployment, and no further refinement is needed during interaction with the black-box model. This characteristic makes offline attacks more stealthy and query-efficient, although their success is heavily dependent on surrogate fidelity, model similarity, and the transferability of adversarial examples [DON], [ZHO], [SHI].

In contrast, score-based query attacks operate entirely in inference-time (online) settings. These methods do not rely on a surrogate model but instead iteratively query the target model to extract real-valued outputs—such as logits or confidence scores—which are used to estimate gradient directions or optimize perturbations via zeroth-order or evolutionary strategies [ILY], [CHE1]. Representative approaches such as NES [ILY], SPSA [SPA], NAttack [LIY], SignHunter [ALD], SimBA [GUO], and AutoZOOM [PCH], [TUC] exemplify this paradigm. NES, SPSA, and NAttack estimate or approximate the gradient using randomized input sampling: NES employs stochastic sampling with Gaussian noise, SPSA perturbs the input in paired random directions to estimate the directional derivative, while NAttack generates adversarial examples by modeling perturbations as samples from a probabilistic (typically Gaussian) distribution and optimizing its parameters via natural gradient descent rather than directly modifying input values. SignHunter estimates only the *sign* of the gradient, rather than its full magnitude, by performing coordinate-wise binary input flips and observing whether each change increases or decreases the model’s loss, drastically reducing query complexity while enabling sign-based adversarial optimization even with only loss information. SimBA generates adversarial examples through a greedy, score-based search that perturbs one input coordinate at a time (e.g., pixel or frequency component), retaining only those changes that reduce confidence in the correct class, which makes it query-efficient and effective, particularly in compact or Fourier-based spaces. AutoZOOM improves query efficiency by shifting optimization from the original high-dimensional input space to a low-dimensional latent space learned via an autoencoder, where gradient estimation and perturbation updates are applied, and then decoded back to perform model queries, significantly reducing query complexity and enhancing scalability to high-resolution inputs. These

attacks construct adversarial perturbations dynamically through repeated querying, offering high flexibility and model-agnostic applicability, but are often computationally expensive, query-intensive, and more susceptible to detection or rate limitation in practical systems.

Decision-based attacks represent an even more restricted inference-time category, as they only rely on final class labels, without access to probability scores or confidence values. These attacks must explore the decision boundary through iterative perturbation and refinement, often requiring a high number of queries to converge, particularly in high-dimensional spaces [AND], [NMA], [CHE2]. Despite their limited observability, decision-based methods remain highly practical for adversarial scenarios targeting commercial APIs that expose only hard-label outputs.

Hybrid strategies, including transfer–query combinations, distribute computational effort across both offline and online phases. They typically begin with adversarial examples generated using a surrogate model and subsequently refine them through targeted querying of the target model [DIN], [CAI], [CHS]. Notable examples include Subspace [GUY], P-RGF [CHS], GFCS [LOR], and BASES [CAI], which enhance surrogate-based attacks by incorporating score-aware gradient refinement. These attacks restrict optimization to a low-dimensional subspace, often aligned with influential directions from the surrogate, to enable efficient gradient estimation without full-dimensional querying. This subspace-aware approach allows perturbations to focus on impactful regions of the input, improving query efficiency, attack success rate, and stealth while reducing noise and enhancing transferability.

TREMBA [HUA] and CGA [FEN] extend this idea by operating in the latent space of generative models such as autoencoders or GANs. Instead of directly manipulating high-dimensional inputs, they learn a distribution over latent perturbations and refine it using score-based feedback from the target model. This latent optimization enables query-efficient adversarial exploration while constraining perturbations to the data manifold, thus enhancing realism, stealth, and convergence. AdvFlow [MOH] further advances hybrid querying by utilizing normalizing flow models to directly learn an invertible mapping from clean inputs to adversarial examples. Unlike surrogate-based approaches, AdvFlow trains a conditional generative model using score-based queries from the target, enabling end-to-end adversarial example generation in purely black-box settings without relying on surrogate alignment.

Additional refinements have introduced adaptive mechanisms for query-efficient online optimization. For instance, real-time adaptive generation and decision boundary approximation [YIN] progressively infers the target model's decision boundary by iteratively probing with small, strategically chosen perturbations. As the attack advances, it adaptively adjusts both direction and magnitude, homing in on minimal perturbations that cause misclassification, making it suitable for fine-grained, online, or streaming systems. Similarly, enhanced gradient estimation techniques [DUJ] improve gradient approximation through adaptive sampling, variance reduction, and incorporation of prior surrogate or query-based information, thus reducing query complexity while maintaining stability and accuracy in high-dimensional settings.

These hybrid methods offer a strong balance between transferability, adaptability, and query efficiency, although often involving greater computational overhead due to model training, subspace construction, or distributional optimization. Nevertheless, they represent a scalable trade-off suitable for constrained, monitored, or real-world environments where query budgets, detection risks, or model access are limited.

6.7 Grey box attacks

The term Grey-box attack, also known as Grey-box attack, refers to attacks in which the adversary has a partial knowledge about the system or the deployed defenses. In most real-life cases, the adversary cannot obtain white-box access, but they may obtain partial information such as some features of the classifier, labels, or in the case of DNN, the results of the hidden layers.

Anthi *et al.* present in [ANT] an inference phase grey box attack which considers an insider attacker that has admin access privileges to a factory’s communication network systems and targets the ML model of the factory’s IDS. Specifically, the attacker has no knowledge of the target model, but they have access to the full dataset and knowledge of features. Although, the attacker has no knowledge of the target model, by using another model they can approximate samples that can cause the model to misclassify due to the transferability of adversarial samples across machine learning models. Consequently, here the attacker has grey-box level of knowledge for the system and uses a transfer attack. The attacker uses a JSMA [ANT] to generate the adversarial samples used in the attack while a pre-trained Multi-Layer Perceptron (MLP) was used as the underlying model for the generation. The JSMA method was chosen because, compared to FGSM, facilitates more realistic and finer-grained AML attacks, as adversaries can define both the percentage of feature to perturb and the amount of perturbation to be included in the generation of adversarial samples. The adversarial examples generated by the JSMA in the context of this attack can be transferred to Random Forest or J48 ML models.

The article [APR] discusses an inference phase evasion query attack deployed against a NIDS that employs machine learning classifiers to identify botnet activities. The attacker’s knowledge is that network communications are monitored by an ML-based NIDS, but they are not aware of what kind of model is integrated in the detector. However, the assume that NIDS model is trained using a dataset containing samples generated by the same or similar malware variant deployed on the infected machines. To avoid detection the attacker uses a targeted exploratory integrity attack [BIG-2] which they perform by inserting tiny modifications in the communication between the bot and its CnC server.

The paper authored by Sidi *et al.* [SID] describes MaskDGA, an inference time evasion attack that uses adversarial learning to modify algorithmically generated domain (AGD) names to evade detection from inline DGA classifiers which detect botnets and the AGDs they belong. The purpose of this attack is to modify AGD names, so they are misclassified as benign by DGA classifiers. It is worth noting that the attacker does not need to possess any knowledge about the DGA classifier’s architecture or its parameters and cannot acquire its outputs, but it must have access to public datasets of AGDs to train the model. The evasion technique comprises of the setup and the attack phase. The setup phase’s goal is to produce a substitute DGA classification model trained by the adversary. Afterwards, the adversary applies the substitute model on the AGD names to gain insight on how to modify them, so they became less detectable by the substitute model. Once they are undetectable, based on the concept of transferability, the modified AGD names will become less detectable by the target models. To train the substitute model on adversarial samples the attacker uses Jacobian-based dataset augmentation. In the attack phase, the adversary and bots use the substitute model to generate domain names through which they will communicate safely while evading detection.

Jo *et al.* in [JUN] present a grey-box adversarial attack against Semi-Supervised (GAAS) Learning models in computer

vision. The attack is performed during the model’s inference phase. Specifically, this attack leverages the publicly available parts of semi-supervised learning model’s training datasets (the initial labeled set) to create a surrogate model. Then adversarial examples, which are transferable, are generated using standard techniques on that surrogate and consequently attack the final semi-supervised learning model in a grey-box setting in a variety of CV related tasks. To generate adversarial examples, the authors use the FGSM method along with the Momentum Iterative FGSM (MIFGSM) and PGD methods. By including on-step and iterative methods to generate adversarial examples, this attack becomes more versatile and effective since it can be deployed against a variety of semi-supervised learning scenarios.

Another grey-box attack is presented by Chiramal *et al.* in [CHI] which targets transformer-based NLP models that are used for text classification and sentiment analysis. Specifically, this attack exploits the fact that, according to the authors, explainable AI removes the black-box nature of AI systems by providing explanations. Thus, it allows adversaries and users to discover existing vulnerabilities within the system. In short, this work proposes an algorithm that exploits the advantages of Explainable AI frameworks and build on adversarial attack on AI models. The core idea of the attack is employing explainable AI techniques to identify which words in an input text affect the most the model’s classification decision. To this end, the attack replaces words with synonyms or with other words to imperceptibly change the targeted text. The attack targets transformer-based NLP models such as bidirectional encoder representations from transformers (BERT) and produces examples that are transferable to other variants of BERT [DEV]. Consequently, this attack becomes a grey-box transfer and query attack. Considering that in this attack the attacker first generates the malicious inputs and submits them to the model after the training phase then this is an inference time attack.

The authors of [QUD] propose a new *gradient-free, grey-box, incremental attack*, aimed specifically at the training phase of neural networks. This method subtly corrupts data structures used to store training instances between epochs, thereby achieving its high-risk impact. The attack’s potency stems from its focus on internal data structures that often go unnoticed by professionals. Specifically, the attacker aims to corrupt the training process by perturbing the training instances between training epochs to optimize a corrupted poisoned discriminant function *F’* which produces faulty probability distributions over the possible output classes. The attack is performed in multiple rounds, and the poisoning function replaces the pixels that fall within a selected area known as the patch with faulty pixels to corrupt the image presentation and lead to a mistaken training process. This attack occurs during the model’s training phase.

The authors of [YWA] proposed a grey-box attack in CLMs known as the membership inference attack. In grey-box *membership inference attacks*, adversaries lack access to the internal representations of the target CLMs, making it difficult to accurately model their behavior. To address this challenge, a common strategy is to use the shadow model training approach. This method involves training shadow models to replicate the behavior of the target CLMs. Since the adversary has full knowledge of the training data for these shadow models, they can use it to extract membership information and train an attack model. The attack model leverages the behavioral similarities between the shadow models and the victim model to infer membership in the victim's training data. The success of this approach is largely dependent on how closely the shadow models mirror the behavior of the victim model.

Torca et al. Presented the N-Pixels grey-box attack in [TOR] against computer vision models and is designed to attack CNNs during the inference phase to force the model make erroneous classifications of images. In N-Pixels the attacker

leverages partial information about the model’s architecture and introduces minimal perturbations in the most important sections of the images examined by the model. In the context of this paper “the term most important section” of the image refers to the part of the image that influences the most the classifier’s decision.

6.8 Categorization based on attacker capability

Another important dimension in our taxonomy is the classification of adversarial attacks based on attacker capability, which refers to the extent of control or influence an adversary has over different components of the ML pipeline, including the data, the model, and its training or inference procedures. This taxonomy reflects the practical feasibility and potential impact of attacks and is crucial for understanding real-world threat models.

This capability-based view is complementary to attacker knowledge taxonomies (e.g., white-box, black-box, grey-box). While knowledge refers to what the attacker knows (e.g., parameters, architecture), capability concerns what the attacker can do, such as injecting data, modifying the model, or crafting inputs. For example, a white-box attacker typically has both high knowledge and high capability, whereas a black-box attacker may have low knowledge and limited capability, such as only being able to query model outputs.

Attacks are often divided into two broad classes based on capability, causative attacks, which occur during training and aim to poison the model by modifying the data or learning process and exploratory attacks, which occur post-training and involve probing or manipulating the model without altering its internal state. These two classes strongly correlate with training-time and inference-time attacks, respectively, though they differ slightly in emphasis. Below, we analyze two key categories that fall under this capability-based taxonomy.

6.9 Causative Attacks

This type of attack represents a class of adversarial strategies that target the *training phase* of ML models. The goal of such attacks is to corrupt the learning process by injecting malicious or misleading data into the training set, thereby inducing the model to learn incorrect patterns, biased representations, or exploitable vulnerabilities. These attacks are highly potent because they influence the model from the ground up, compromising its foundational understanding of the task. Causative attacks include data poisoning, where adversaries insert specially crafted inputs with incorrect labels or misleading features, and backdoor attacks, in which the model learns to associate hidden triggers with specific outputs [BUR]. The consequence is a model that performs poorly on unseen data or that behaves maliciously when triggered, resulting in increased false positives, false negatives, or even model subversion. Because causative attacks manipulate the data before the model is even deployed, they are particularly dangerous in scenarios where training data comes from untrusted or decentralized sources [SIH, NEL].

6.10 Exploratory attacks

Carried out *after the model has been trained*, during the inference or deployment phase [SET, IBI]. The adversary does not interfere with the training data but instead probes the model’s behavior to extract information or to craft malicious inputs that result in erroneous outputs. This category includes *evasion attacks*, where adversarial examples are constructed by subtly perturbing input data to bypass detection or classification systems [BIG, PIT]. Another form is *model inversion attacks*, in which an attacker leverages access to model predictions or confidence scores to reconstruct sensitive training data, such as biometric profiles or personal attributes [HEZ, WUX]. Exploratory attacks can occur in both white-box and black-box settings, depending on the level of access the adversary has to the model internals.

These attacks are often stealthy, require minimal interaction, and can compromise the confidentiality and integrity of systems even when the training process is secure, making them a critical concern in deployed AI services.

6.11 Bridging Timing and Capability Taxonomies

As introduced earlier, adversarial attacks can be categorized by their timing (training-time vs inference-time) and by the capability of the attacker (causative vs. exploratory). These two perspectives often align but emphasize different aspects. Causative attacks correspond closely to training-time attacks, as they aim to compromise the learning process itself, typically through techniques like data poisoning, label flipping, or backdoor insertion [ANT]. The term "causative" places emphasis on the influence over model behavior, whereas "training-time" emphasizes the temporal phase of the intervention.

Similarly, exploratory attacks generally align with inference-time attacks [ZAR]. These attacks occur after training and involve probing, querying, or subtly manipulating inputs to extract sensitive information or evade detection. Examples include evasion attacks, model extraction, and membership inference. Some exploratory attacks may even exploit vulnerabilities implanted during training, such as in Trojan attacks. Within this category, more nuanced subtypes exist:

- *Transfer-based attacks*, which rely on surrogate models in a grey-box setting.
- *Query-based attacks*, which use model output probabilities in black-box scenarios.
- *Decision-based attacks*, which only observe predicted labels.
- *Hybrid attacks*, which combine multiple strategies to maximize stealth or effectiveness.

These refinements serve as a bridge between timing- and capability-based taxonomies, helping us understand how, when, and to what extent adversaries can compromise machine learning systems across different threat models.

6.12 Granularity and Constraints of Data Manipulation

Beyond the broad classification of attacks based on their timing or adversary capability, it is equally important to examine how adversarial manipulation operates at the data level. This perspective introduces an orthogonal but highly informative dimension of categorization: granularity (what part of the data is manipulated) and constraints (how that manipulation is performed).

Data manipulation refers to the deliberate alteration of input data by an adversary with the goal of misleading or subverting a machine learning model [LIA, JAG]. It is a core mechanism underlying many adversarial attacks, particularly in causative scenarios such as data poisoning. By carefully manipulating specific aspects of the training or test data, attackers can alter the model’s decision boundaries or behavior, either to degrade general accuracy or to achieve specific targeted misclassifications.

Two key axes define this manipulation:

- *Feature vs. Sample Targeting*: this dimension distinguishes whether the attacker targets individual features (e.g., pixels, tokens, attributes) or manipulates entire samples. Feature manipulation is common in evasion attacks, where slight perturbations in key dimensions can fool a classifier. Sample-level manipulation includes

the injection of poisoned or backdoored instances in training data, or the crafting of highly misleading but valid test samples.

- *Manipulation Constraints*: adversaries typically face constraints that limit how much they can alter data. These may be mathematical (e.g., bounded L_∞ , L_2 or L_0[14] norms), semantic (preserving grammatical correctness or class meaning), or perceptual (imperceptibility to human observers). Such constraints are crucial to maintaining the stealthiness and feasibility of attacks, especially in real-world domains like vision, language, or cybersecurity.

Understanding these dimensions enhances our ability to analyze attacks not just by *when* or *who* conducts them, but by *how* adversarial control is operationalized at the data level. These manipulation-based categories can thus be applied across both training-time and inference-time attacks, and across both white-box and black-box threat models.

Building upon the previous discussion of data manipulation as a core adversarial capability, we now explore its three principal forms: feature-level manipulation, sample-level manipulation, and manipulation under constraints. These categories help operationalize the mechanics of adversarial influence, offering a detailed perspective on how attacks are crafted, regardless of when they occur (training vs inference) or what the attacker knows (white/black-box).

- *Feature Manipulation:* feature manipulation refers to the process by which an adversary subtly alters specific input features to deceive a machine learning model while keeping the overall input distribution largely unchanged. This strategy is especially potent because even minimal perturbations in key features can lead to significant misclassifications, particularly in high-dimensional spaces such as image or audio recognition tasks [DAS]. For instance, in computer vision, an attacker may modify pixel values imperceptibly to human observers but cause a model to misclassify a traffic sign or a face. In structured data, this could involve tweaking fields such as income or age in a fraud detection dataset. Feature manipulation is frequently used in evasion attacks, where the goal is to bypass a decision boundary without raising alarms.

- *Sample Manipulation:* sample manipulation encompasses adversarial actions where entire data instances are created, modified, or removed to influence the learning process or the decision boundaries of a model. In poisoning scenarios, attackers inject harmful training examples crafted to be indistinguishable from legitimate data, that skew the learned representations [CHA]. A more targeted form appears in backdoor attacks, where a trigger pattern embedded in training samples causes the model to consistently misclassify such inputs at test time [DUA]. Sample manipulation may also include data omission, where representative instances are selectively removed to introduce bias or fragility. These actions affect not only accuracy, but also fairness, generalization, and robustness, especially in imbalanced or decentralized data settings.

- *Constraints on Manipulation*: in real-world attacks, adversaries must often operate under constraints that limit

[14] The L_0 norm is not a true mathematical norm, but a measure of sparsity. It counts the number of non-zero components in a vector, rather than measuring their magnitude. As such, it reflects how many entries are active or significant. The L_0 measure is widely used in compressed sensing and sparse signal reconstruction, although its direct optimization is computationally difficult due to its non-convex and discontinuous nature.

the nature or magnitude of allowable data changes. These constraints serve both to preserve the realism of inputs and to evade detection by humans or automated defenses [KOT, DAI]. In image-based attacks, for example, perturbations are commonly bounded by L_∞ [15] or L_2 [16] norms to remain imperceptible [BHA, GUE]. In text-based domains, changes must preserve semantic, syntactic, and grammatical validity [CRO, NIM]. In cybersecurity, packet modifications must respect protocol integrity. These constraints increase the difficulty of the attack but also make it more plausible, emphasizing the need for sophisticated optimization methods.

These three perspectives, feature vs. sample granularity, and manipulation constraints, form a cross-cutting lens that enriches our earlier taxonomies. Regardless of whether an attack occurs during training or inference, or whether the adversary operates with full or limited knowledge, it is ultimately these fine-grained manipulations that operationalize the attack. As such, these dimensions provide the tactical level of control that complements the broader strategic classifications based on timing, capability, and knowledge, offering a more holistic view of the adversarial threat landscape.

6.13 Categorization based on type

Another foundational axis in our taxonomy is the categorization of attacks based on type, which reflects the fundamental objective and mode of influence that the adversary pursues. This classification broadly distinguishes between *poisoning attacks* and evasion attacks, two of the most prominent and widely studied forms of adversarial behavior. Poisoning attacks aim to corrupt the learning process by injecting malicious data during training, thereby degrading model performance or embedding hidden behaviors. In contrast, *evasion attacks* are performed after the model has been trained, with the goal of crafting inputs that are misclassified at inference time without altering the model itself. This type-based distinction complements the previously introduced timing and capability-based taxonomies, while offering a conceptual separation grounded in adversarial intent, whether the attacker seeks to compromise the model’s internal representation or merely to bypass it at runtime.

6.14 Poisoning Attacks

Poisoning attacks target the AI model’s learning process during its training phase so that in can be trained in a way that suites the adversary’s goal. Poisoning attacks can be separated into the three following categories [BOU]:

- *Training data modification*: one of the most direct forms of poisoning attacks, wherein the adversary either alters, removes, or injects malicious instances into the training dataset to compromise the learning process. This manipulation may include subtle corruption of existing samples, deletion of crucial data that would aid

[15] The L_∞ norm, also known as the maximum or supremum norm, measures the largest absolute value among the components of a vector. In the case of functions, it corresponds to the largest value the function attains (more precisely, the essential supremum). Unlike other norms that capture average magnitude, the L_∞ norm reflects the worst-case or peak value.

[16] The L_2 norm, also known as the *Euclidean* norm, measures the overall magnitude of a vector by taking the square root of the sum of the squares of its components. For functions, it corresponds to the square root of the integral of the squared function over its domain. Unlike the L_∞ norm, which captures only the maximum value, the L_2 norm reflects the total energy or average strength of the signal or vector.

generalization, or the insertion of carefully crafted adversarial examples. The attacker’s goal is to steer the model toward learning biased or misleading patterns, which may degrade performance globally or introduce specific vulnerabilities. For example, poisoning a spam classifier’s training set with non-spam emails labelled as spam can lead to elevated false negatives during deployment. Since many AI systems rely on crowdsourced, user-generated, or unverified data, this type of attack is particularly relevant in open or decentralized data collection settings [MBA].

- *Label manipulation or Label Flipping*: involves deliberately altering the class labels of training samples in supervised learning environments. By mismatching labels with their corresponding inputs, the adversary aims to inject semantic confusion into the model’s learning. This attack can severely impact classification performance, especially when applied to critical classes (e.g., flipping benign and malicious labels in malware detection or flipping “stop sign” and “yield” in traffic sign classification). Label flipping can be implemented uniformly (random flips across classes) or strategically (targeting specific class pairs to degrade certain predictions). As this technique targets the integrity of the labelled data, it requires access to both features and labels and is most impactful when models are trained without robust data validation or noise-resilient learning algorithms [BIG].
- *Input Feature Manipulation*: targets the internal representations and statistical properties learned during training by modifying the features of the input data. This type of poisoning does not necessarily alter labels but instead injects inputs with misleading or distorted attributes, aiming to cause misgeneralization or force the model to rely on irrelevant or spurious correlations. For example, an attacker may slightly alter the pixel distributions of handwritten digits in a digit recognition dataset or manipulate metadata fields in structured data to encode adversarial biases. This technique is particularly dangerous because it can lead to models learning incorrect feature importances or latent patterns, resulting in misclassification or susceptibility to test-time attacks. Input feature manipulation can also be combined with label flipping for compounded effect, and is often used in stealthier, model-targeted poisoning scenarios [JAG].

Poisoning attacks, while distinct in their objective to compromise the training process, exhibit conceptual ties to other adversarial strategies, most notably backdoor attacks, and later, as we will discuss, evasion attacks. In poisoning, the adversary manipulates the training data to corrupt the model’s learned decision boundaries, leading to widespread or targeted misclassification at inference. Backdoor attacks can be seen as a specialized form of poisoning, where carefully crafted training inputs implant hidden behaviors that are triggered only under specific conditions post-deployment. In both cases, the attacker exerts control during training to predetermine how the model will behave at inference. Although differing in scope, with poisoning aimed at widespread degradation of model performance, while backdoor attacks enable targeted, trigger-based exploitation, both attack types challenge the fundamental assumption that training data is clean and trustworthy. These relationships illustrate how poisoning-based techniques serve as a bridge between causative manipulation and the more reactive, inference-time threats we explore next, such as evasion attacks.

6.15 Evasion attacks

Evasion attacks [BIG] form a core class of adversarial techniques that operate exclusively during the inference phase, aiming to bypass or deceive a trained model without altering its internal parameters or training data. What sets these

attacks apart is their reactive nature: rather than influencing how the model learns, the attacker exploits weaknesses in the model's learned decision boundaries by crafting specially modified inputs that trigger incorrect predictions. These perturbations are often subtle, bounded by perceptual or mathematical constraints, but carefully optimized to induce misclassification. Unlike training-time attacks, evasion techniques typically assume no control over the learning process and often operate under black-box or limited-access conditions, making them highly practical in real-world deployment scenarios.

The defining feature of evasion attacks is their post-training exploitative strategy, which contrasts with the constructive interference seen in causative attacks such as poisoning or backdoors. However, despite this temporal and operational distinction, evasion and training-time attacks share a conceptual link: both capitalize on the fragility of the model’s generalization and its sensitivity to high-dimensional inputs. Viewed through the lens of our broader taxonomies, evasion attacks clearly belong to the exploratory category in the capability-based framework, are inherently aligned with inference-time attacks in the timing-based taxonomy and often fall within the black-box or grey-box regimes under the knowledge-based classification. Furthermore, from the manipulation-based perspective, evasion attacks typically involve fine-grained feature perturbations subject to explicit norm-based constraints, emphasizing stealth and minimal detectability. These cross-cutting characteristics reinforce how the taxonomy based on attack type, poisoning versus evasion, not only provides a clear conceptual division based on adversarial goals, but also integrates meaningfully with the other dimensions we have established, offering a comprehensive and multi-layered understanding of adversarial threats in machine learning systems.

Figure 2 illustrates a hierarchical, tree-structured taxonomy of the adversarial attacks summarized in Table 4. The taxonomy organizes the attacks primarily according to the adversary’s level of knowledge, distinguishing among white-box, grey-box, and black-box threat models. This knowledge level constitutes the conceptual root of the tree, reflecting the degree of access the adversary has to the target model’s architecture, parameters, and training data. From this root, the taxonomy branches into application domains, namely Computer Vision, NLP, CLM, Cybersecurity, and IoT. Each domain node further subdivides according to the adversary’s knowledge setting, resulting in a clear separation of attacks based on both domain-specific characteristics and threat assumptions. The terminal nodes (leaves) of the tree correspond to concrete adversarial attack methods. For each attack, an explicit tag is provided to indicate the attack stage: “I” denotes inference-time (evasion) attacks, while “T” denotes training-time (poisoning) attacks. In instances where multiple attacks share identical domain and knowledge characteristics, they are grouped under the same leaf and listed sequentially, separated by commas, to improve readability and conserve visual space.

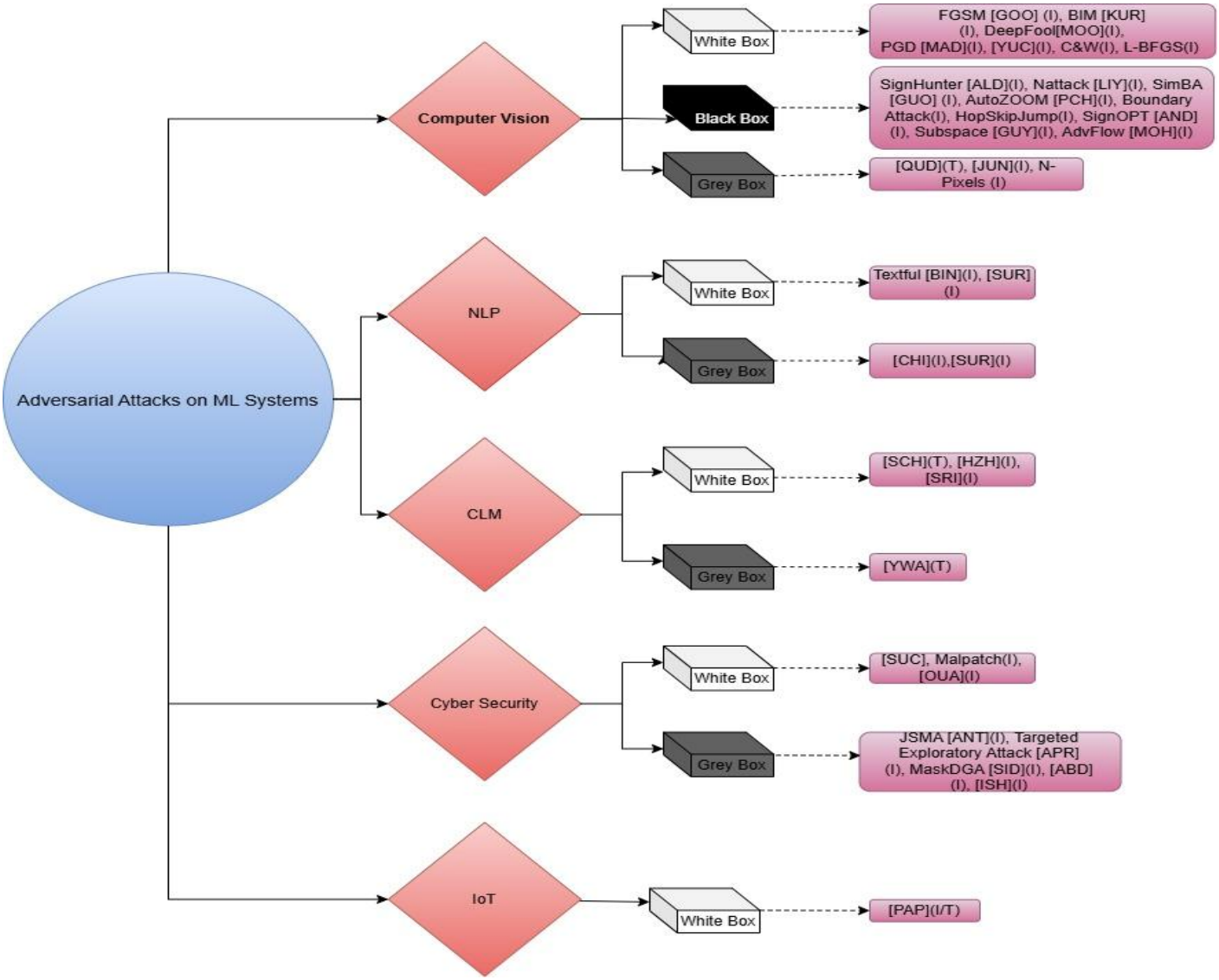

Figure 2: Tree taxonomy of adversarial attacks on ML systems.

7 Impact of adversarial examples

Adversarial Examples (AEs) have had a profound impact on the understanding and trustworthiness of deep learning models, particularly by exposing critical vulnerabilities in systems once considered highly accurate and robust. These subtle and imperceptible perturbations to input data can cause models to make confident yet incorrect predictions, revealing that state-of-the-art neural networks may be easily manipulated. This fragility raises serious concerns about the deployment of AI in security-sensitive domains, such as facial recognition, autonomous vehicles, medical diagnosis, and financial systems, where the cost of misclassification can be catastrophic. The existence of adversarial examples challenges the assumption that performance on benchmark datasets directly translates to reliability in real-world environments. Furthermore, their transferability across different models suggests that even black-box systems, which hide their internal workings, are not immune to such threats. As a result, adversarial attacks have prompted a growing body of research in adversarial robustness, secure model training, and certified defenses, while simultaneously influencing policymakers and practitioners to rethink the safe deployment and governance of AI technologies.

The paper by Vatian *et al.* [VAT] offers a compelling real-world exploration of how AEs can disrupt deep learning systems in high-stakes domains, specifically in medical imaging. Unlike typical adversarial attacks that involve artificially crafted perturbations, this work raises a critical question: *Can real medical images, due to inherent technological noise, unintentionally act as natural adversarial examples when processed by neural networks?* The study investigates two high-tech medical imaging modalities: Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), and demonstrates that, even in the absence of intentional manipulation, some real patient scans can cause significant misclassification when analyzed using CNN-based systems. This highlights a crucial insight. AEs are not merely synthetic artifacts generated by malicious actors but may also emerge naturally due to imperfections in the imaging pipeline, noise, or model oversensitivity. Quantitatively, the study reveals that misclassification rates without any defense mechanisms were non-trivial, 200 per 10,000 samples for CT and 285 per 10,000 samples were misclassified for MRI scans. These numbers drastically improved with the integration of simple yet effective defense strategies such as bounded ReLU activations, Gaussian noise data augmentation, and adversarial training, reducing errors to as low as 12 and 15 respectively. These findings underscore the practical relevance of adversarial robustness techniques even in standard clinical workflows, where errors can have life-or-death consequences. Moreover, the authors observe that adversarial training not only improves classification robustness but also alters the nature of adversarial perturbations, making them visibly distinguishable to human clinicians. This implies a potential synergy between machine learning defenses and human-in-the-loop decision making, reinforcing the importance of interpretability and redundancy in AI-assisted medical diagnosis.

In the broader context of adversarial example research, this paper exemplifies how AEs expose critical reliability gaps in AI systems deployed in sensitive environments. It bridges theoretical work on perturbation vulnerability with real-world implications for clinical trust, patient safety, and ethical deployment. The study also illustrates that even unintentional AEs can hinder the interpretability and utility of high-tech diagnostic tools, emphasizing the need for robust model design and continual evaluation in real-world conditions.

The work by Devabhakthini *et al.* [DEA] addresses an often-overlooked consequence of adversarial examples: their impact on model explainability, particularly in NLP tasks. While much of the adversarial ML literature focuses on performance degradation, this paper highlights a critical secondary effect: how adversarial perturbations can distort the interpretability of machine learning models, thereby undermining trust in AI systems even when outputs appear plausible. The authors investigate adversarial attacks on a text classification model and examine how these attacks affect post hoc explanation mechanisms, such as word importance and feature attribution. By introducing small, strategically crafted changes to input text (e.g., synonym substitution, word reordering, or inserting misleading terms), the adversarial examples not only cause misclassification but also manipulate the internal attention or attribution maps used for explaining the model’s decisions. This distortion is particularly dangerous in high-stakes domains such as healthcare triage systems, content moderation, or legal document analysis, where stakeholders rely on explainable AI (XAI) methods to interpret, verify, or challenge automated decisions. The study reveals that adversarial inputs can cause the model to assign high importance to irrelevant or misleading tokens, misguiding users who interpret the explanations in good faith. Through empirical analysis, the paper demonstrates that explanations before and after the adversarial attack differ significantly, indicating that adversarial perturbations can not only mislead the model but also its human users. This dual vulnerability calls into question the robustness of explainable ML pipelines, where models are assumed to offer transparency and accountability. By focusing on the intersection of adversarial robustness and

interpretability, this work contributes to a growing body of research that challenges the assumption that explainability is a static or reliable property of AI systems. It also suggests that defenses against adversarial examples must extend beyond accuracy preservation to include stability of explanations under adversarial conditions. Ultimately, the paper underscores the need for robust interpretability techniques, capable of resisting adversarial manipulation, and reinforces the broader concern that adversarial examples can erode both the functional and epistemic trust in machine learning systems.

8 Impact of adversarial AI attacks

While the phenomenon of adversarial examples has traditionally been explored from a model-centric and research-oriented perspective highlighting the intrinsic vulnerabilities of deep learning models, the concept of *AAI attacks* reframes this discussion within a threat-oriented and security-critical context [QIU, KON]. Here, the central concern is not merely the theoretical existence of model weaknesses, but their deliberate exploitation by malicious actors to subvert, deceive, or disable AI systems deployed in real-world, high-stakes environments [KAV].

Computational perception systems, spanning image recognition, speech processing, augmented reality, sentiment analysis, and natural language understanding, have revolutionized industry and society. Yet, these same systems are increasingly exposed to adversarial manipulation. The threat landscape now encompasses not only traditional cyberattack vectors but also machine learning-specific exploits that weaponize the mathematical and structural properties of AI models. These include evasion attacks (misleading models at inference time), poisoning attacks (compromising models during training), model extraction attacks (reconstructing models via queries), and membership inference attacks (revealing training data).

The consequences of such attacks are profound. Unlike conventional software flaws that can often be addressed with patches or updates, adversarial AI vulnerabilities stem from the very learning processes that define these systems, such as non-linearity, overfitting, and poor generalization. In practice, this creates a persistent and evolving risk that can undermine system performance, trust, and security.

AAI attacks have already manifested across multiple domains, where the cost of failure can be substantial. Below, we analyze the specific impact of AAI across various sectors.

Safety-Critical Systems: autonomous vehicles, drones, and other safety-critical AI applications are highly sensitive to input perturbations. Even minimal changes, such as altered traffic signs, can cause misclassification by perception systems. The divergence between human and machine interpretation in such environments elevates the risk of catastrophic outcomes. Facial and voice recognition systems, widely deployed in surveillance and access control, are similarly susceptible to adversarial bypasses.

Real-world cases further illustrate these risks. In 2017, researchers demonstrated that Tesla’s Autopilot system could be deceived with subtle visual alterations, leading to traffic misinterpretation [TES]. A 2023 Waymo incident [WAY] highlighted how ambiguous sensor inputs can impair perception, even without a confirmed adversarial cause. In such contexts, adversarial actors may employ camouflage, sensor spoofing, or environmental manipulation to compromise autonomous operations.

Deepfakes and Disinformation: the advent of generative models has enabled the creation of sophisticated deepfakes capable of evading detection by both humans and automated systems. Adversarial attacks on deepfake detectors threaten public trust, enabling misinformation campaigns and the erosion of institutional credibility. These attacks are not merely technical nuisances; they represent a growing vector for information warfare and social manipulation.

Healthcare Systems: AI in healthcare, particularly in diagnostic imaging, is vulnerable to adversarial perturbations that can lead to misdiagnoses. Malicious modifications to scans can manipulate model outputs, posing life-threatening risks. Additionally, adversaries can exfiltrate or poison model parameters, undermining both accuracy and patient privacy. Given the high stakes in medical decision-making, these vulnerabilities demand stringent verification and oversight mechanisms.

Financial Systems and Algorithmic Trading: adversarial attacks on financial systems, such as fraud detectors or trading algorithms, can exploit model blind spots by subtly altering input features. The resulting failures can lead to financial fraud, market manipulation, and systemic instability. In one illustrative case, adversarially crafted transactions could evade detection, leading to privacy breaches, legal repercussions, and reputational damage.

Voice Assistants and Smart Devices: inaudible adversarial commands targeting voice assistants demonstrate how AI can be manipulated in ways imperceptible to humans. These covert instructions can activate devices, leak sensitive data, or execute unauthorized operations. This attack vector necessitates advances in signal filtering, biometric validation, and anomaly detection.

Surveillance and Biometric Security: adversarial inputs can defeat biometric systems by exploiting facial, gait, or fingerprint recognition vulnerabilities. The combination of social engineering and adversarial design can allow attackers to evade detection or impersonate authorized users. Cross-modal validation and human oversight are essential to mitigate such threats.

Military Systems: the military domain presents unique challenges for adversarial AI defense. AI-enabled platforms, such as drones and battlefield robots, operate in contested, resource-constrained environments where physical and digital compromise is more likely. The adoption of edge computing further increases the attack surface, as models and data reside directly on field-deployed devices. Military AI systems often rely on shared datasets and pretrained models across branches. While efficient, this practice introduces systemic risk: a single compromised dataset can propagate vulnerabilities across multiple applications. Outsourcing and insufficient oversight in data preparation further elevate the likelihood of model poisoning or unauthorized access. Additionally, military operations are characterized by limited diagnostics and real-time monitoring. This makes adversarial reconnaissance and latent model compromise difficult to detect, especially when attackers delay activation of their attacks. Accordingly, robustness, explainability, and adversarial resilience must be treated as foundational design principles in defense-related AI [RAD].

Beyond current deployments, AAI threatens also the future integrity of rapidly developing AI-based technologies [COM]:

Cross-Domain Vulnerabilities: AAI can compromise adjacent technologies in cybersecurity (e.g., malware detectors), genomics (e.g., protein structure models), and advanced manufacturing (e.g., generative design systems). In each case,

adversarial inputs can distort outputs or subvert system intent, threatening operational safety, data reliability, and downstream processes. In the anticipated Internet of Intelligent Things (IoIT), the interconnected, heterogeneous nature of intelligent devices makes them particularly prone to adversarial manipulation. As systems update asynchronously and interact dynamically, adversarial actors may exploit inconsistencies in assumptions, behaviors, or trust boundaries.

Advanced Persistent Threats (APTs) and Insider Risks: AAI empowers sophisticated threat actors to enhance every phase of an APT campaign: reconnaissance, payload generation, delivery, exploitation, and command-and-control. AI can be used to craft context-aware phishing emails, simulate social personas, or bypass filters via generative mimicry. Insiders with privileged access pose additional risks. They may inject poisoned data, exfiltrate sensitive model parameters, or manipulate inference results. AAI also enables personalized social engineering attacks by mining behavioral and biometric data from publicly available sources. These threats highlight the need for comprehensive internal monitoring and AI-aware cybersecurity frameworks.

Satellite Imagery and Remote Sensing: satellite systems are susceptible to adversarial manipulation at multiple points in the imagery processing pipeline. From obfuscating physical targets to misleading anomaly detection algorithms, attackers can use AAI to evade observation or generate false intelligence. As commercial satellite data becomes more accessible, adversarial threats are likely to proliferate. Multi-modal sensor fusion and redundant detection layers offer potential avenues for resilience.

Foundation Models: Scale, Emergence, and Homogenization: foundation models, such as large-scale, pre-trained systems used across diverse tasks, are particularly vulnerable to AAI due to their scale and opacity. The vastness of their training data makes validation nearly impossible, opening the door to data poisoning and inference attacks. Malicious content, if scraped into the training corpus, can compromise model integrity and propagate errors downstream. The generative and emergent behaviors of foundation models further complicate the threat landscape. These systems may hallucinate, fabricate, or align outputs with adversarial prompts, making them vehicles for misinformation and manipulation. Moreover, as foundation models are reused across sectors, a single point of compromise can introduce widespread vulnerabilities.

Adversarial AI represents a multifaceted and deeply systemic threat to the trustworthy deployment of machine learning technologies. From battlefield robots to smart cities, and from healthcare diagnostics to financial modeling, the scope of impact is both broad and profound. As attackers innovate and adapt, the arms race between defense and exploitation intensifies. As a result, these attacks can be exploited to compromise a wide range of targets, sometimes indirectly, including industrial control systems, environmental monitoring and control systems, AI-based malware detection mechanisms, and the daily operations of SMEs. These affected areas align with the use cases identified in D2.2 Specifications & Business Cases. Addressing these risks requires an interdisciplinary response encompassing robust model design, secure data handling, adversarial training, threat modeling, and policy regulation. Technical innovation alone is insufficient. A resilient future for AI demands a holistic approach that bridges machine learning, cybersecurity, and institutional governance.

9 Discussion

The current taxonomy organized adversarial attacks on AI and ML models using the adversary’s knowledge (white-box, grey-box, and black-box) of the model and indicates a negative relationship between the attacker’s knowledge and the attacks practicality. Specifically, as attacker’s knowledge decreases, practical relevance increases, while the gap in achievable attack success rate narrows dramatically.

White-box attacks (FGSM, BIM, PGD, C&W, DeepFool, etc.) constitute the theoretical maximum limit of the adversary’s capabilities and the de-facto standard for robustness evaluation. Their nature yields a high probability of success, but they rely on unrealistic full access that rarely exists outside insider or open-source scenarios.

Grey-box attacks offer an effective balance between practical realism and adversarial strength. These attacks rely on limited yet meaningful auxiliary information, such as publicly available datasets, knowledge of the model family, or access to explainability outputs, while still operating without full transparency into the underlying system. By leveraging this partial insight and the well-documented transferability of adversarial examples, grey-box approaches remain both feasible in real-world scenarios and substantially more potent than purely black-box methods.

Black-box attacks are considered the most realistic and operationally significant threat model for deployed AI systems. In practical deployment environments adversaries typically lack access to the internal architecture, parameters, or training data of the underlying models. Instead, they interact with the system solely through exposed interfaces, querying the model and observing its outputs. This constraint makes black-box attack strategies highly relevant, as they reflect the actual access patterns and limitations an attacker is likely to face, thereby posing a credible and pressing security challenge for real-world AI deployments.

To effectively protect an AI/ML model from adversarial attacks, multiple factors must be considered. First, evaluating robustness solely under white-box conditions is no longer sufficient, as these scenarios assume that the attacker has full knowledge of the model. This an assumption that rarely holds in real-world environments. In practice, adversaries usually operate under grey-box or black-box conditions, where their access to internal model details is limited or entirely restricted. Nonetheless, attacks in these settings have become highly effective due to techniques such as transferability, surrogate modeling, and low-query hybrid strategies. Consequently, relying exclusively on white-box evaluations can create a false sense of security and leave critical vulnerabilities unaddressed. Comprehensive robustness assessments must therefore include grey-box and black-box threats to reflect realistic adversarial capabilities.

Another critical consideration in designing defences is the aspect of transferability. Transferability becomes a significant weakness because adversarial examples crafted for one model often remain effective against other models, even when those models differ in architecture, training data, or internal parameters. This allows an attacker to train or approximate a surrogate model using only limited information and then generate adversarial inputs on that surrogate. Since these inputs frequently “transfer” to the actual target model, attackers can circumvent defenses that rely on obscurity or restricted access. As a result, transferability enables highly potent attacks even under grey-box and black-box conditions, underscoring the need for defences that explicitly account for this vulnerability.

This taxonomy examined 35 attacks in total against AI, ML, and DL models as well as neural networks, which are summarized in Table 4. The most frequently targeted application domain of the examined attacks is Computer Vision, totaling 19 out of 35 attacks. The main reason behind the occurrence of attacks with this orientation is that Computer

Vision is utilized in a variety of everyday critical applications and sectors, such as facial recognition in security applications, healthcare [GAO] and manufacturing [ISL]. In particular, in manufacturing, computer vision helps in quality control and automation, while in agriculture [SUM], it assists in monitoring crops and in pest detection. Computer vision is also leveraged in autonomous vehicles and particularly in navigation and obstacle avoidance operations [KAN]. Another vital application domain of computer vision is medical image analysis, in which case an AI/ML model or neural network automates the detection of a disease and helps a physician determine if and what treatment a person should receive.

Therefore, it becomes apparent that through these attacks, an attacker not only seeks to disrupt the services and operations of these sectors as well as obtain unauthorized access for their own gain, but also to endanger and harm lives. It is worth noting that half of the examined attacks on computer vision do not require knowledge of the level since the attacker can deceive the system by tampering with the object’s appearance in some cases.

Next in occurrence in our taxonomy are the adversarial attacks oriented against cybersecurity systems, which are 8 in total and require the attacker to have partial or full knowledge of the targeted system. These attacks mostly target neural networks to prevent them from identifying malware, phishing, anomalous user behavior and malicious network traffic patterns. If successfully executed, such attacks can impact a variety of systems and applications used in critical infrastructures and disrupt their smooth operation. Affected critical sectors and infrastructure can range from smart grids, healthcare, telecommunications, and transportation. The number of attacks against cybersecurity systems is low compared to computer vision due to the fact that partial or full knowledge of the system is required by the attacker, which, in most cases, the adversary does not possess and requires insider access. The same cause justifies the low number of attacks in our taxonomy oriented against the domains of Code Language Models, Natural Language Processing and IoT.

Only four of the examined attacks in the current taxonomy occur during the training phase, with the main reason behind this low number being that they require tampering with the training dataset. This is challenging as it demands at least partial access to the model, control over data sources, and access to the data set. This kind of access is very rare and difficult to obtain in secure, deployed systems in which training data is monitored, and provided by trusted entities.

All in all, attacks, such as the ones described in this taxonomy, not only may disrupt the daily activities of a society’s critical infrastructure and endanger human lives but also affect the public’s trust in AI systems. Consequently, when such attacks are successfully carried out, they indirectly form an obstacle to the wide adoption of AI and ML models and systems.

Attack	Application Domain	Phase of attack (training or inference)	Knowledge level	Attack Type (Poisoning or other)	Type of attacked model
FGSM [GOO]	Computer Vision	Inference	White	Evasion	DNN
BIM [KUR]	Computer Vision	Inference	White	Evasion	ML
DeepFool [MOO]	Computer Vision	Inference	White	Evasion	DNN
PGD [MAD]	Computer Vision	Inference	White	Evasion	DL
Hu et al. [YUC]	Computer Vision	Inference	White	Evasion	DL
Suciu et al. [SUC]	Cybersecurity	Inference	White	Evasion	CNN
Carlini & Wagner [CAR]	Computer Vision	Inference	White	Evasion	DNN
L-BFGS	Computer Vision	Inference	White	Evasion	DL
Textfool [BIN]	Natural Language Processing	Inference	White	Evasion	Convolutional & Recurrent Neural Networks
[SUR]	Natural Language Processing	Inference	White	Evasion	ML
[SCH]	Code Language Model	Training	White	Poisoning	DNN
[HZH]	Code Language Model	Inference	White	Evasion	DL
[SRI2]	Code Language Model	Inference	White	Evasion	ML

MalPatch [ZHA]	Cybersecurity	Inference	White	Evasion	DNN
[PAP2]	IoT	Training & Inference	White	Poisoning and Evasion	ML & DL
[OUA]	Cybersecurity	Inference	White	Evasion	Recurrent Neural Network, Deep Neural Network, Convolutional Neural Network
SignHunter [ALD]	Computer Vision	Inference	Black	Evasion	DNN
Nattack [LIY]	Computer Vision	Inference	Black	Evasion	DNN
SimBA [GUO]	Computer Vision	Inference	Black	Evasion	Neural Network
AutoZOOM [PCH]	Computer Vision	Inference	Black	Evasion	DNN
Boundary Attack	Computer Vision	Inference	Black	Evasion	DNN
HopSkipJump	Computer Vision	Inference	Black	Evasion	DNN
SignOPT [AND]	Computer Vision	Inference	Black	Evasion	DNN
Subspace [GUY]	Computer Vision	Inference	Black	Evasion	DNN
AdvFlow [MOH]	Computer Vision	Inference	Black	Evasion	DL
N-Pixels	Computer Vision	Inference	Grey	Evasion	CNN

JSMA [ANT]	Cybersecurity	Inference	Grey	Evasion	ML
Targeted Exploratory Integrity Attack [APR]	Cybersecurity	Inference	Grey	Evasion	ML & DL
MaskDGA [SID]	Cybersecurity	Inference	Grey	Evasion	Deep Convolutional Neural Network & Recurrent Neural Network
Chiramal [CHI]	Natural Language Processing	Inference	Grey	Evasion	ML
[ABD]	Cybersecurity	Inference	Grey	Evasion	DNN
[ISH]	Cybersecurity	Inference	Grey	Evasion	DNN
[QUD]	Computer Vision	Training	Grey	Poisoning	DNN
[JUN]	Computer Vision	Inference	Grey	Evasion	DNN, Semi-Supervised Learning
[YWA]	Code Language Model	Training	Grey/Black (Hybrid approach)	Poisoning	Recurrent Neural Network

10 Conclusions

This deliverable presents a comprehensive taxonomy of adversarial attacks targeting ML and AI models across key application domains, including NLP, Malware Detection, Cybersecurity, CV, IoT, and CLM. These domains underpin critical sectors of everyday life, such as autonomous vehicles, public transportation, industrial control systems, and medical diagnostic technologies. As a result, adversaries capable of executing successful attacks against these systems pose serious threats, ranging from endangering human lives and disrupting essential services to compromising the integrity of data and reputation of organizations.

The taxonomy introduced in this deliverable classifies adversarial attacks based on a multidimensional methodology that integrates application domain, the attacker's level of knowledge about the target model, and the phase at which the attack occurs (training or inference). The attacker's knowledge influences the likelihood of an attack's success, while the timing of the attack provides insight into its potential impact. By combining these factors, the proposed taxonomy offers a nuanced understanding of the threat landscape and enables more accurate assessments of the risks adversarial attacks pose to critical infrastructure and their impact.

References

[ABD] Abdullah Al-Dujaili, Alex Huang, Erik Hemberg, and Una-May O’Reilly. 2018. Adversarial Deep Learning for Robust Detection of Binary Encoded Malware. In Proc. of the 2018 IEEE Security and PrivacyWorkshops (SPW2018). Francisco, CA, USA, 76–82.

[ALD] Al-Dujaili, Abdullah, and Una-May O'Reilly. "Sign bits are all you need for black-box attacks." *International conference on learning representations*. 2020.

[AND] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, “Square attack: a query-efficient black-box adversarial attack via random search,” in European conference on computer vision. Springer, 2020, pp. 484–501.

[ANT] Anthi, E., Williams, L., Rhode, M., Burnap, P., & Wedgbury, A. (2021). Adversarial attacks on machine learning cybersecurity defences in industrial control systems. Journal of Information Security and Applications, 58, 102717.

[APR] Apruzzese, G., Andreolini, M., Marchetti, M., Colacino, V. G., & Russo, G. (2020). AppCon: Mitigating evasion attacks to ML cyber detectors. Symmetry, 12(4), 653.

[BAR] Barreno, Marco, et al. "The security of machine learning." Machine learning 81 (2010): 121-148.

[BHA] Bhattad, Anand, et al. "Unrestricted adversarial examples via semantic manipulation." *arXiv preprint arXiv:1904.06347* (2019).

[BID] Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O’Brien, K., Hallahan, E., ... & Van Der Wal, O. (2023, July). Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning* (pp. 2397-2430). PMLR.

[BIG] B. Biggio, B. Nelson, P. Laskov, Support vector machines under adversarial label noise, in: Asian Conference on Machine Learning, PMLR, 2011, pp. 97–112.

[BIG2] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., ... & Roli, F. (2013). Evasion attacks against machine learning at test time. In Machine learning and knowledge discovery in databases: European conference, ECML pKDD 2013, prague, czech Republic, September 23-27, 2013, proceedings, part III 13 (pp. 387-402). Springer Berlin Heidelberg.

[BIG3] Biggio, Battista, and Fabio Roli. "Wild patterns: Ten years after the rise of adversarial machine learning." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018.

[BIN] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep Text Classification Can be Fooled. arXiv preprint arXiv:1704.08006 (2017).

[BOU] Bountakas, P., Zarras, A., Lekidis, A., & Xenakis, C. (2023). Defense strategies for adversarial machine learning: A survey. *Computer Science Review*, *49*, 100573.

[BUR] Burkard, Cody, and Brent Lagesse. "Analysis of causative attacks against svms learning from data streams." Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics. 2017.

[CAI] Cai, C. Song, S. Krishnamurthy, A. Roy-Chowdhury, and S. Asif, “Blackbox attacks via surrogate ensemble search,” in Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 5348–5362.

[CAR] Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp) (pp. 39-57). Ieee.

[CHA] Chakraborty, Anirban, et al. "A survey on adversarial attacks and defences." *CAAI Transactions on Intelligence Technology* 6.1 (2021): 25-45.

[CHA2] Chaganti, K. Adversarial Attacks on AI-driven Cybersecurity Systems: A Taxonomy and Defense Strategies. Authorea Preprints.

[CHE1] J. Chen, M. I. Jordan, and M. J. Wainwright, “Hopskipjumpattack: A query-efficient decision-based attack,” in IEEE Symposium on Security and Privacy, 2020.

[CHE2] M. Cheng, S. Singh, P.-Y. Chen, S. Liu, and C.-J. Hsieh, “Sign-opt: A query-efficient hard-label adversarial attack,” in International Conference on Learning Representations (ICLR), 2020.

[CHE3] W. Chen, Z. Zhang, X. Hu, and B. Wu, “Boosting decision-based black- box adversarial attacks with random sign flip,” in European Conference on Computer Vision (ECCV), 2020.

[CHI] Chiramal, E., & Kai, K. S. B. (2025). A Grey-box Text Attack Framework using Explainable AI. arXiv preprint arXiv:2503.08226.

[CHS] Cheng, Shuyu, et al. "Improving black-box adversarial attacks with a transfer-based prior." *Advances in neural information processing systems* 32 (2019).

[CHV] Chistyakov, Alexander, and Alexey Andreev. "AI under Attack." *How to secure machine learning in security systems, Kaspersky Threat Research, dated Aug* 27 (2019).

CMU Software Engineering Institute, Carnegie Mellon University, 2017, https://www.sei.cmu.edu/documents/503/1996_019_001_496172.pdf#page=123 , Online, Accessed on 11-12-2025

[COM] Comiter, Marcus. "Attacking artificial intelligence." Belfer Center Paper 8 (2019): 2019-08.

[CRO] Crothers, Evan N., Nathalie Japkowicz, and Herna L. Viktor. "Machine-generated text: A comprehensive survey of threat models and detection methods." *IEEE Access* 11 (2023): 70977-71002.

[D2.2] AIAS, D2.2: Specifications & Business cases, https://aias-project.eu/wp-content/uploads/2025/07/AIAS_D2.2.pdf , online, Accessed on 26-11-2025

[DAI] Dai, Aobotao, et al. "When Data Manipulation Meets Attack Goals: An In-depth Survey of Attacks for VLMs." *arXiv preprint arXiv:2502.06390* (2025).

[DAS] Dasgupta, Dipankar, and Kishor Datta Gupta. "Dual-filtering (DF) schemes for learning systems to prevent adversarial attacks." *Complex & Intelligent Systems* 9.4 (2023): 3717-3738.

[DEA] Devabhakthini, Prathyusha, et al. "Analyzing the impact of adversarial examples on explainable machine learning." *arXiv preprint arXiv:2307.08327* (2023).

[DEM] Demontis, Ambra, et al. "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks." *28th USENIX security symposium (USENIX security 19)*. 2019.

[DEV] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).

[DIN] K. Ding, X. Liu, W. Niu, T. Hu, Y. Wang, and X. Zhang, “A low-query black-box adversarial attack based on transferability,” Knowledge-Based Systems, vol. 226, p. 107102, 2021.

[DON] Y. Dong, T. Pang, H. Su, and J. Zhu, “Evading defenses to transferable adversarial examples by translation-invariant attacks,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4307–4316.

[DUA] Duan, Mingxing, et al. "A novel multi-sample generation method for adversarial attacks." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 18.4 (2022): 1-21.

[DUJ] Du, Jiawei, et al. "Query-efficient meta attack to deep neural networks." *arXiv preprint arXiv:1906.02398* (2019).

[DTH] D. Thanh Tran, H. Su Le, and J.-H. Huh, “Building an automatic irrigation Fertilization system for smart farm in greenhouse,” IEEE Trans. Consum. Electron., vol. 70, no. 2, pp. 4685–4698, May 2024.

[FEN] Feng, Yan, et al. "Boosting black-box attack with partially transferred conditional adversarial distribution." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[FEY] Feng, Hua, et al. "A Comparative Analysis of White Box and Gray Box Adversarial Attacks to Natural Language Processing Systems." 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024). Atlantis Press, 2024.

[GAO] Gao, J., Yang, Y., Lin, P., & Park, D. S. (2018). Computer vision in healthcare applications. Journal of healthcare engineering, 2018, 5157020.

[GOL] M. Goldblum et al., "Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 2, pp. 1563-1580, 1 Feb. 2023, doi: 10.1109/TPAMI.2022.3162397.

[GOO] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[GUE] Guesmi, Amira, et al. "Physical adversarial attacks for camera-based smart systems: Current trends, categorization, applications, research challenges, and future outlook." *IEEE Access* 11 (2023): 109617-109668.

[GUO] Guo, Chuan, et al. "Simple black-box adversarial attacks." *International conference on machine learning*. PMLR, 2019.

[GUY] Guo, Yiwen, Ziang Yan, and Changshui Zhang. "Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks." *Advances in Neural Information Processing Systems* 32 (2019).

[HAS] Hassija, Vikas, et al. "Interpreting black-box models: a review on explainable artificial intelligence." Cognitive Computation 16.1 (2024): 45-74.

[HEZ] He, Zecheng, Tianwei Zhang, and Ruby B. Lee. "Model inversion attacks against collaborative inference." *Proceedings of the 35th Annual Computer Security Applications Conference*. 2019.

[HIR] Hirano, H., & Takemoto, K. (2020). Simple iterative method for generating targeted universal adversarial perturbations. Algorithms, 13(11), 268.

[HUA] Huang, Zhichao, and Tong Zhang. "Black-box adversarial attack with transferable model-based embedding." *arXiv preprint arXiv:1911.07140* (2019).

[HUA2] Huang, Ling, et al. "Adversarial machine learning." Proceedings of the 4th ACM workshop on Security and artificial intelligence. 2011.

[HZH] H. Zhang, Z. Fu, G. Li, L. Ma, Z. Zhao, H. Yang, Y. Sun, Y. Liu, and Z. Jin, “Towards robustness of deep program processing models—detection, estimation, and enhancement,” ACM Trans. Softw. Eng. Methodol., vol. 31, pp. 1–40, 2022.

[IBI] Ibitoye, Olakunle, et al. "The Threat of Adversarial Attacks on Machine Learning in Network Security--A Survey." *arXiv preprint arXiv:1911.02621* (2019).

[ILY] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, “Black-box adversarial attacks with limited queries and information,” in International Conference on Machine Learning (ICML), 2018.

[ISH] Ishai Rosenberg, Asaf Shabtai, Lior Rokach, and Yuval Elovici. 2017. Generic Black-Box End-to-End Attack against RNNs and Other API Calls Based Malware Classifiers. arXiv preprint arXiv:1707.05970 (2017).

ISL Islam, M. R., Zamil, M. Z. H., Rayed, M. E., Kabir, M. M., Mridha, M. F., Nishimura, S., & Shin, J. (2024). Deep learning and computer vision techniques for enhanced quality control in manufacturing processes. *IEEE Access*.

[JAG] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, B. Li, Manipulating machine learning: Poisoning attacks and countermeasures for regression learning, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, pp. 19–35.

[JAG] Jagielski, Matthew, et al. "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning." *2018 IEEE symposium on security and privacy (SP)*. IEEE, 2018.

[JAK] J. Akhter, R. Hazra, A. Mihovska, and R. Prasad, “A novel resource sharing scheme for vehicular communication in 5G cellular networks for smart cities,” IEEE Trans. Consum. Electron., vol. 70, no. 3, pp. 5848–5856, Aug. 2024, doi: 10.1109/TCE.2024.3392435.

[JIA] Jiang, Y., Yin, G., Yuan, Y., & Da, Q. (2021). Project gradient descent adversarial attack against multisource remote sensing image scene classification. Security and Communication Networks, 2021(1), 6663028.

[JIJ] Jiang, J., Wang, F., Shen, J., Kim, S., & Kim, S. (2024). A survey on large language models for code generation. arXiv preprint arXiv:2406.00515.

[JUN] Jo, J., Kim, J., & Suh, Y.-J. (2024). Exploring Public Data Vulnerabilities in Semi-Supervised Learning Models through Gray-box Adversarial Attack. Electronics, 13(5), 940. https://doi.org/10.3390/electronics13050940

[KAL] Kalin, Josh, et al. "Black box to white box: Discover model characteristics based on strategic probing." *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*. IEEE, 2020.

[KAN] Kanchana, B., Peiris, R., Perera, D., Jayasinghe, D., & Kasthurirathna, D. (2021, December). Computer vision for autonomous driving. In 2021 3rd international conference on advancements in computing (ICAC) (pp. 175-180). IEEE.

[KAV] Kaviani, Sara, Ki Jin Han, and Insoo Sohn. "Adversarial attacks and defenses on AI in medical imaging informatics: A survey." *Expert Systems with Applications* 198 (2022): 116815.

[KAW] Kawamoto, Y., Miyake, K., Konishi, K., & Oiwa, Y. (2023). Threats, Vulnerabilities, and Controls of Machine Learning Based Systems: A Survey and Taxonomy. *arXiv preprint arXiv:2301.07474*.

[KHA] Khazane, H.; Ridouani, M.; Salahdine, F.; Kaabouch, N. A Holistic Review of Machine Learning Adversarial Attacks in IoT Networks. Future Internet 2024, 16, 32. https://doi.org/10.3390/fi16010032

[KON] Kong, Zixiao, et al. "A survey on adversarial attack in the age of artificial intelligence." Wireless Communications and Mobile Computing 2021.1 (2021): 4907754.

[KOT] Kothari, Nupur, et al. "Finding protocol manipulation attacks." *Proceedings of the ACM SIGCOMM 2011 Conference*. 2011.

[KUR] Kurakin, A., Goodfellow, I. J., & Bengio, S. (2018). Adversarial examples in the physical world. In Artificial intelligence safety and security (pp. 99-112). Chapman and Hall/CRC.

[LIA] Liao, Cong, et al. "Server-based manipulation attacks against machine learning models." *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018.

[LIC] Li, C., Wang, H., Yao, W., & Jiang, T. (2024). Adversarial attacks in computer vision: a survey. *Journal of Membrane Computing*, *6*(2), 130-147.

[LIY] Li, Yandong, et al. "Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks." *International conference on machine learning*. PMLR, 2019.

[LOR] Lord, Nicholas A., Romain Mueller, and Luca Bertinetto. "Attacking deep networks with surrogate-based adversarial black-box methods is easy." *arXiv preprint arXiv:2203.08725* (2022).

[MAD] Mądry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. stat, 1050(9).

[MAX] Ma, Xiao, and Wu-Jun Li. "Grey-box adversarial attack on communication in multi-agent reinforcement learning." Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems. 2023.

[MBA] M. Barreno, B. Nelson, R. Sears, A.D. Joseph, J.D. Tygar, Can machine learning be secure? in: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, 2006, pp. 16–25.

[MOH] Mohaghegh Dolatabadi, Hadi, Sarah Erfani, and Christopher Leckie. "Advflow: Inconspicuous black-box adversarial attacks using normalizing flows." *Advances in Neural Information Processing Systems* 33 (2020): 15871-15884.

[MOO] Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2574-2582).

[MUN] Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., & Roli, F. (2017, November). Towards poisoning of deep learning algorithms with back-gradient optimization. In Proceedings of the 10th ACM workshop on artificial intelligence and security (pp. 27-38).

[NAQ] Naqvi, S. M. A., Shabaz, M., Khan, M. A., & Hassan, S. I. (2023). Adversarial attacks on visual objects using the fast gradient sign method. Journal of Grid Computing, 21(4), 52.

[NEL] Nelson, Blaine, et al. "Exploiting machine learning to subvert your spam filter." LEET 8.1-9 (2008): 16-17.

[NIM] Ni, Mingze. *Unmasking Vulnerabilities: Adversarial Attacks via Word-Level Manipulation on NLP Models*. Diss. University of Technology Sydney (Australia), 2024.

[NIST] NIST AI 100-2 E2025, Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations, NIST, https://csrc.nist.gov/pubs/ai/100/2/e2025/final?Offer=ab_ss_reeng_plt_var , Accessed online on 14-11-2025

[NMA] C. Ma, L. Chen, and J. Yong, “Simulating unknown target models for query-efficient black-box attacks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’21), 2021, pp. 11 830–11 839.

[OUA] Ouazza, H., Khennou, F., & Abdellaoui, A. (2025). Adversarial Retraining and White-Box Attacks for Robust Malware Detection. 2025 13th International Symposium on Digital Forensics and Security (ISDFS), 1-6.

[PAP] 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 ACM Asia Conference on Computer and Communications Security (CCS’17), 2017.

[PAP2] Papadopoulos, P., Thornewill von Essen, O., Pitropakis, N., Chrysoulas, C., Mylonas, A., & Buchanan, W. J. (2021). Launching Adversarial Attacks against Network Intrusion Detection Systems for IoT. Journal of Cybersecurity and Privacy, 1(2), 252-273. https://doi.org/10.3390/jcp1020014

[PAP3] Papernot, Nicolas, et al. "Sok: Security and privacy in machine learning." 2018 IEEE European symposium on security and privacy (EuroS&P). IEEE, 2018.

[PCH] Chen, Pin-Yu, et al. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models." *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 2017.

[PER] Perera, Manoj Madushanka, et al. "A Survey of the State-of-the-Art in Conversational Question Answering Systems." arXiv preprint arXiv:2509.05716 (2025).

[PIS] Pispa, A., & Halunen, K. (2024). A comprehensive artificial intelligence vulnerability taxonomy. In *European Conference on Cyber Warfare and Security* (Vol. 23, pp. 379-387).

[PIT] Pitropakis, Nikolaos, et al. "A taxonomy and survey of attacks against machine learning." *Computer Science Review* 34 (2019): 100199.

[QIN] Y. Qin, Y. Xiong, J. Yi, and C.-J. Hsieh, “Training meta-surrogate model for transferable adversarial attack,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 8, 2023, pp. 9516–9524.

[QIU] Qiu, Shilin, et al. "Review of artificial intelligence adversarial attack and defense technologies." *Applied Sciences* 9.5 (2019): 909.

[QUD] Al-Qudah, R., Aloqaily, M., Ouni, B., Guizani, M., & Lestable, T. (2023, May). An incremental gray-box physical adversarial attack on neural network training. In *ICC 2023-IEEE International Conference on Communications* (pp. 45-50). IEEE.

[RAD] Radanliev, Petar, and Omar Santos. "Adversarial attacks can deceive AI systems, leading to misclassification or incorrect decisions." ACM Computing Surveys (2023).

[ROD] Rodrigues, Ricardo N., Lee Luan Ling, and Venu Govindaraju. "Robustness of multimodal biometric fusion methods against spoof attacks." *Journal of Visual Languages & Computing* 20.3 (2009): 169-179.

[SCH] R. Schuster, C. Song, E. Tromer, and V. Shmatikov, “You autocomplete

me: poisoning vulnerabilities in neural code completion,” in Proc. USENIX Security, 2021, pp. 1559–1575.

[SET] Sethi, Tegjyot Singh, and Mehmed Kantardzic. "Data driven exploratory attacks on black box classifiers in adversarial domains." *Neurocomputing* 289 (2018): 129-143.

[SHA] Shayea, G. G., Zabil, M. H. M., Habeeb, M. A., Khaleel, Y. L., & Albahri, A. S. (2025). Strategies for protection against adversarial attacks in AI models: An in-depth review. *Journal of Intelligent Systems*, *34*(1), 20240277.

[SHI] Y. Shi, S. Wang, and Y. Han, “Curls & whey: Boosting black-box adversarial attacks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’19), 2019, pp. 6512–6520.

[SID] Sidi, L., Nadler, A., & Shabtai, A. (2020). MaskDGA: An evasion attack against DGA classifiers and adversarial defenses. IEEE access, 8, 161580-161592.

[SIH] Sihag, Saurabh, and Ali Tajer. "Secure estimation under causative attacks." IEEE Transactions on Information Theory 66.8 (2020): 5145-5166.

[SPA] Spall, James C. "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation." *IEEE transactions on automatic control* 37.3 (1992): 332-341.

[SRI2] S. Srikant, S. Liu, T. Mitrovska, S. Y. Chang, Q. Fan, G. Zhang, and U.-M. O’Reilly, “Generating adversarial computer programs using optimized obfuscations,” in Proc. ICLR, 2021, pp. 1–9.

[SUC] Suciu, O., Coull, S. E., & Johns, J. (2019, May). Exploring adversarial examples in malware detection. In 2019 IEEE Security and Privacy Workshops (SPW) (pp. 8-14). IEEE.

[SUM] Sumaira Ghazal, Arslan Munir, Waqar S. Qureshi, Computer vision in smart agriculture and precision farming: Techniques and applications, Artificial Intelligence in Agriculture, Volume 13, 2024, Pages 64-83, ISSN 2589-7217, https://doi.org/10.1016/j.aiia.2024.06.004.

[SUR] Suranjana Samanta and Sameep Mehta. 2018. Generating Adversarial Text Samples. In Proc. of the 40th European Conference on IR Research (ECIR 2018). Grenoble, France, 744–749.

[SZE] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2013, arXiv preprint arXiv:1312.6199.

[TES] MIT Technology Review, Hackers trick a Tesla into veering into the wrong lane, 01-04-2019, https://www.technologyreview.com/2019/04/01/65915/hackers-trick-teslas-autopilot-into-veering-towards-oncoming-traffic/ , Last accessed on 13-06-2025

[TOR] Salvatore Della Torca, Valentina Casola, and Simone Izzo. 2025. N-Pixels: a Novel Grey-Box Adversarial Attack for Fooling Convolutional Neural Networks. In Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25). Association for Computing Machinery, New York, NY, USA, 1539–1547. https://doi.org/10.1145/3672608.3707944

[TUC] Tu, Chun-Chen, et al. "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

[VAT] Vatian, Aleksandra, et al. "Impact of adversarial examples on the efficiency of interpretation and use of information from high-tech medical images." *2019 24th Conference of Open Innovations Association (FRUCT)*. IEEE, 2019.

[WAY] AI Incident Database, “Incident 640: Waymo Software Flaw Leads to Double Collision with Tow Truck”, 11-12-2023 https://incidentdatabase.ai/cite/640/ Last accessed on 13-06-2025

[WAN] Wang, S., Ko, R. K., Bai, G., Dong, N., Choi, T., & Zhang, Y. (2023). Evasion attack and defense on machine learning models in cyber-physical systems: A survey. *IEEE communications surveys & tutorials*, *26*(2), 930-966.

[WIE] Wierstra, Daan, et al. "Natural evolution strategies." *The Journal of Machine Learning Research* 15.1 (2014): 949-980.

[WUC] Wu, Chenwang, et al. "Genetic algorithm with multiple fitness functions for generating adversarial examples." *2021 IEEE Congress on evolutionary computation (CEC)*. IEEE, 2021.

[WUX] Wu, Xi, et al. "A methodology for formalizing model-inversion attacks." *2016 IEEE 29th computer security foundations symposium (CSF)*. IEEE, 2016.

[XWA] X. Wang and Y. Wu, “Fog-assisted Internet of Medical Things for smart healthcare,” IEEE Trans. Consum. Electron., vol. 69, no. 3, pp. 391–399, Aug. 2023.

[YAN] Yang, Y., Fan, H., Lin, C., Li, Q., Zhao, Z., Shen, C., & Guan, X. (2024). A Survey on Adversarial Machine Learning for Code Data: Realistic Threats, Countermeasures, and Interpretations. *arXiv preprint arXiv:2411.07597*.

[YIN] Yin, Fei, et al. "Generalizable black-box adversarial attack with meta learning." *IEEE transactions on pattern analysis and machine intelligence* 46.3 (2023): 1804-1818.

[YUC] Hu, Y. C. T., Kung, B. H., Tan, D. S., Chen, J. C., Hua, K. L., & Cheng, W. H. (2021). Naturalistic physical adversarial patch for object detectors. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 7848-7857).

[YWA] Y. Wan, G. Wan, S. Zhang, H. Zhang, Y. Sui, P. Zhou, H. Jin, and L. Sun, “Does your neural code completion model use my code? A membership inference approach,” ArXiv:2404.14296, 2024.

[ZAR] Zaremba, W., et al. "Trading inference-time compute for adversarial robustness, 2025." *URL https://cdn. openai. com/papers/trading-inference-time-compute-for-adversarial-robustness-20250121_1. pdf*.

[ZHA] Zhan, D., Duan, Y., Hu, Y., Li, W., Guo, S., & Pan, Z. (2023). MalPatch: Evading DNN-based malware detection with adversarial patches. *IEEE Transactions on Information Forensics and Security*, *19*, 1183-1198.

[ZHO] Zhong and W. Deng, “Towards transferable adversarial attack against deep face recognition,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1452–1466, 2021.