Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Hacking intelligence: Mapping the anatomy of adversarial threats in artificial intelligence with MITRE ATLAS☆

Nikolaos Sachpelidis-Brozos[a], Efklidis Katsaros[a], Panagiotis Radoglou-Grammatikis[a,*], Georgios Kalitsios[a], Antonios Sarigiannidis[a], George Calin Seritan[b], Ilias Politis[c], Christos Xenakis[d], Sotirios Goudos[e], Panagiotis Sarigiannidis[f]

[a] *K3Y Ltd, Studentski District, Vitosha Quarter, Bl. 9, Sofia, 1700, Bulgaria*
[b] *Faculty of Electrical Engineering, University "Politehnica" of Bucharest, Bucharest, 061071, Romania*
[c] *Industrial Systems Institute, Research Center "ATHENA", Patras Science Park Building, Platani, Patras, 26504, Greece*
[d] *Secure Systems Laboratory, Department of Digital Systems, University of Piraeus, 80 Karaoli & Dimitriou, Piraeus, 18534, Greece*
[e] *School of Physics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece*
[f] *Department of Electrical and Computer Engineering, University of Western Macedonia, Active Urban Planning Zone, Kozani, 50100, Greece*

HIGHLIGHTS

- Structured analysis using MITRE ATLAS to classify AI threats across six attack families, mapping tactics and techniques.
- Systematic analysis of 63 methods across domains, evaluating theory, threat models, datasets, and results.
- Actionable defense strategies mapping 24 mitigation mechanisms to MITRE ATLAS techniques as a practical guide for practitioners.
- Synthesis of findings highlighting research gaps, future directions, and proposed enhancements to the MITRE ATLAS framework.

ARTICLE INFO

Keywords:
Adversarial attacks
Artificial intelligence
Cybersecurity
MITRE ATLAS

ABSTRACT

Recent advancements in digital technologies and the integration of artificial intelligence (AI) with software systems have introduced new challenges in cybersecurity. Traditional frameworks such as MITRE ATT&CK have proven expressive enough for the analysis of software threats, yet they are limited in accommodating the vulnerabilities of ML systems. In response, MITRE ATLAS was developed to extend the threat analysis specifically to AI and machine learning (ML) environments, providing a structured taxonomy for adversarial tactics and techniques attempting to compromise them. In this paper, we extend the conversation by reviewing papers related to adversarial attacks and examining their categorization, their theoretical foundations, and their advancements compared to prior work. Specifically, we analyze a total of 63 papers across the entire AI attack spectrum and further delve into their objectives, threat models, scientific advancements, and evaluation. Our contributions include the first, to-date analysis of attack vectors following the MITRE ATLAS paradigm, a synthesis of recent advancements, and a discussion on the limitations in the current body of knowledge. We hope that our analysis clarifies the present challenges and serves as a foundation for future research towards securing AI systems.

☆ This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101168407 (cPAID). Disclaimer: Funded by the European Union. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

* Corresponding author.

Email addresses: nsachpelidis@k3y.bg (N. Sachpelidis-Brozos), ekatsaros@k3y.bg (E. Katsaros), pradoglou@k3y.bg (P. Radoglou-Grammatikis), gkalitsios@k3y.bg (G. Kalitsios), asarigia@k3y.bg (A. Sarigiannidis), george.seritan@upb.ro (G.C. Seritan), ilpolitis@isi.gr (I. Politis), xenakis@unipi.gr (C. Xenakis), sgoudo@physics.auth.gr (S. Goudos), psarigiannidis@uowm.gr (P. Sarigiannidis).

URL: https://k3ylabs.com/ (N. Sachpelidis-Brozos), https://k3ylabs.com/ (E. Katsaros), https://k3ylabs.com/g (P. Radoglou-Grammatikis), https://k3ylabs.com/ (G. Kalitsios), https://k3ylabs.com/ (A. Sarigiannidis), https://upb.ro/en/faculties/the-faculty-of-electrical-engineering/ (G.C. Seritan), https://www.athenarc.gr/ (I. Politis), https://www.ds.unipi.gr/ (C. Xenakis), https://www.physics.auth.gr/ (S. Goudos), https://ithaca.ece.uowm.gr/ (P. Sarigiannidis).

https://doi.org/10.1016/j.cosrev.2026.100923
Received 28 April 2025; Received in revised form 24 January 2026; Accepted 29 January 2026
Available online 6 February 2026

1. Introduction

The rapid adoption of digital technologies has introduced heavy reliance on interconnected software systems across industries, governments, and societies [136]. From cloud computing to Internet of Things (IoT) devices, these systems support healthcare services and financial transactions. Unfortunately, this dependence on software introduces significant security risks. Even minor flaws in code or design can be exploited by attackers to disrupt services, steal data, or cause harm [173]. To systematize these threats, frameworks like MITRE ATT&CK, short for Adversarial Tactics, Techniques, and Common Knowledge, have been introduced [152]. MITRE ATT&CK provides a standardized taxonomy for detecting and mitigating adversarial behaviors across the cyber kill chain. It categorizes adversarial techniques aimed at compromising software infrastructure such as phishing, privilege escalation, and lateral movement, to help organizations anticipate threats, profile attackers, and defend effectively.

Nowadays, artificial intelligence (AI) and machine learning (ML) are complementing traditional software, introducing new layers of cybersecurity risks [125]. Unlike static software, ML systems learn from data, adapt to new inputs, and operate probabilistically. These characteristics create novel attack surfaces [182]. Malicious actors can exploit these dynamic models by contaminating training sets to insert malicious triggers, [16], manipulating inputs to produce incorrect outputs, [50], or recovering private parameters through reverse-engineering techniques [135]. Far from being theoretical concerns, such attacks have already undermined facial recognition systems, led autonomous vehicles astray, and circumvented fraud detection platforms [170].

Notably, AI systems are vulnerable to both digital and physical attacks, with numerous real-life examples. In one digital attack, an individual used evasion techniques to exploit ID.me's identity verification system in California [164]. The attacker paired stolen personal information with fake driving licenses and selfies of himself wearing wigs. Using these materials, he filed at least 180 fraudulent unemployment claims, stealing over $3.4 million before being arrested. On the physical front, AI-based cyber-physical systems are equally at risk. For example, in June 2019, researchers revealed vulnerabilities in global navigation satellite systems (GNSS) dependent platforms by successfully spoofing the global positioning system (GPS) navigation of a Tesla Model 3 [36]. By manipulating navigation data, the attackers demonstrated how spoofing tactics could impact real-time driving decisions, calling for further research into stronger cybersecurity measures for GPS technologies.

The transition from traditional software to AI-driven systems highlights the need for adaptive security frameworks, [125]. While MITRE ATT&CK, [152] addresses conventional cyber threats targeting software infrastructure, there is still a significant gap regarding the threats targeting the AI landscape. MITRE responded to the growing risks faced by AI and ML systems by creating ATLAS, short for Adversarial Threat Landscape for AI Systems [101]. It is an offshoot of the MITRE ATT&CK framework that focuses on threats unique to AI. MITRE ATLAS highlights how adversarial objectives evolve when targeting ML models. For example, an attacker might exploit a biased model to manipulate loan approvals or weaponize a misclassified image to cause autonomous systems to fail, as discussed in [170].

The MITRE ATLAS framework facilitates the organization of the complex threat landscape of ML-based applications. It is organized hierarchically and provides a structure to classify concepts and knowledge about threats, adversarial tactics, and mitigations. Moreover, it is expressive enough to support in-depth analysis and reasoning about both attacks and defenses. By contextualizing these threats, MITRE ATLAS not only raises awareness but also assists organizations in proactively defending AI systems [89]. This paper explores how the MITRE ATLAS framework provides a roadmap for securing modern technological ecosystems, ensuring resilience against legacy and emerging cyber threats [195].

In this paper, we review the MITRE ATLAS taxonomy and discuss the classification and relevant case studies that motivate it. Thereafter, we carefully select and review 63 research papers relevant to adversarial attacks of various types, which essentially embody as described by the MITRE ATLAS framework. Our contributions are summarized as follows:

- We provide the first-to-date analysis of adversarial attacks through the lens of MITRE ATLAS, systematically mapping threats into six distinct categories: Evasion, Poisoning, Model Extraction, Inference, Model Inversion, and LLM-related attacks. By identifying tactics, objectives, and corresponding techniques, we provide a structured understanding of the evolving threat landscape.
- We conduct a detailed and rigorous analysis of 63 selected research papers covering a wide spectrum of domains and modalities, from traditional Computer Vision (CV) and Natural Language Processing (NLP) to Graph Neural Networks (GNN), and more recently, Large Language Models (LLMs). Our analysis not only categorizes these works but also systematically evaluates their theoretical contributions, threat models, datasets, and experimental outcomes, offering deep insights into the state-of-the-art.
- We introduce a dedicated analysis of defense mechanisms in Section 6, mapping 24 mitigation strategies directly to their respective ATLAS attack techniques. This strengthens the practical relevance of the work by demonstrating how each threat can be countered with effective and actionable defense strategies.
- We distill our analysis in concise limitations of the current literature and discuss them in the context of future research directions. Furthermore, we propose structural improvements to the MITRE ATLAS framework to address novel and emerging attack vectors.

The rest of this paper is organized as follows. Section 2 discusses related surveys on adversarial attacks across different modalities, threat models, and learning paradigms. Section 3 presents the methodological framework. In Section 4 we provide the background and discuss in detail the miter ATLAS tactics, their objectives, and the techniques they include. The techniques are thereafter mapped to diverse attack categories in line with the existing literature, and a total of 63 papers are analyzed in Section 5 to provide an in-depth overview of the attack landscape. Section 6 discusses defense and mitigation methods for handling these attacks. In Section 7, we discuss open research avenues that are yet understudied. Lastly, in Section 8 we revisit and conclude the main parts of this work.

2. Related survey works

In a seminal survey Yuan et al. [182] discuss adversarial attacks at test time, widely termed evasion attacks. They analyze attacks based on the threat model, perturbation and reported benchmarks. Herein, the authors decompose the threat model into four aspects: adversarial falsification, adversary's knowledge, adversarial specificity, and attack frequency. Adversarial falsification includes false positive attacks, where benign inputs are misclassified as malicious, and false negative attacks, where malicious inputs evade detection. Adversary's knowledge distinguishes between white-box attacks, where the attacker knows the model's details, and black-box attacks, where only output information is available. Adversarial specificity differentiates targeted attacks, which force misclassification into a specific category, from non-targeted attacks, which aim for any incorrect classification. Lastly, attack frequency compares one-time attacks, which generate adversarial examples in a single step, with iterative attacks, which refine examples over multiple iterations. Regarding perturbation, attacks are distinguished by whether they seek an individual (sample-specific) or universal (sample-agnostic) perturbation to impact the model. Moreover, they consider whether the perturbation is the optimization objective or a constraint of the problem and lastly, they consider the magnitude. Finally, the attacks are categorized according to the dataset and the model(s) they are evaluated against. The paper goes beyond mere discussion of the methods and considers many applications as well as respective defenses.

Pitropakis et al. [125] discuss adversarial machine learning across three main tasks, i.e., intrusion detection, spam filtering, and visual recognition. The authors categorize the attack phases into preparation and manifestation, the former referring to gathering the intelligence required to prepare the attack plan. The manifestation step involves launching the attack, which depends on the attacker's knowledge, the algorithm used, and whether game-theoretic approaches are employed. The attack can occur in either the training (poisoning) or testing (evasion) phases, targeted or not, and can be the product of an individual attacker or a joint collaboration among multiple colluding attackers. Finally, attacks are categorized by evaluation method (analytical or experimental) and by their impact on performance, measured as drops in classification or clustering accuracy.

Rigaki and Garcia [135] discuss privacy and confidentiality attacks in ML and categorize them into four types. Membership Inference Attacks aim to determine if a specific data sample was part of the training set. These attacks include passive and active variants, as well as auditing approaches from a data owner's perspective. They apply to both supervised models (black-box and white-box) and generative models, such as generative adversarial networks (GANs) and variational autoencoders (VAEs). Reconstruction Attacks, also termed attribute inference or model inversion, seek to recreate training samples or labels either fully or partially. They may also generate class representatives or probable sensitive feature values, such as facial data in classification tasks. Property Inference Attacks extract unintended dataset properties, such as demographic ratios or latent biases, that are unrelated to the model's training task. These attacks target dataset-wide traits or batch-level patterns (e.g., in collaborative learning), with implications for privacy and security. Model Extraction Attacks focus on replicating a target model's behavior via substitute models. These attacks aim for either task accuracy or decision boundary fidelity and often serve as precursors to adversarial or membership inference attacks. These attacks may also recover hyperparameters, architectural details (e.g., activation types, layers), or optimization algorithms, emphasizing efficiency in query usage and model complexity. All types of attacks in the proposed taxonomy are analyzed a) w.r.t. the attacker's knowledge of the system and b) in both the typical centralized scenario and within the federated learning paradigm.

In a more recent work, Fang et al. [45] review techniques on model inversion. Similar to how training data with some principles can derive a model, a model with some principles can derive training data. Again, the main distinction is the attackers knowledge. Specifically, a white-box scenario implies that the attacker has full access to the weights and outputs of the target model, whereas black box access implies access to confidence scores or raw decision outputs. This taxonomy is organized based on two different axes. First, the reconstructed data modality, i.e., whether one attempts to recover image, text, graph or tabular data from a given model. Second, the tasks the model was trained for, i.e., classification, generation, or representation learning. The authors go beyond model inversion attacks and further discuss defenses.

In another survey paper, Oliynyk et al. [113] review model extraction attacks and corresponding defenses. These attacks are categorized based on the adversary's objective, using the stealing objective as the differentiation axis. Usually the main goal is to replicate the model's behavior, which falls into two subcategories: (1) attacks aiming to closely approximate the model's predictions (accuracy) and (2) those attempting to replicate its decision-making process as closely as possible (fidelity). Additionally, some attacks focus on extracting specific model properties, such as the target model's hyperparameters, architecture, or training details. Furthermore, the authors divide adversarial motivation into two main types: (1) those where the attackers try to replicate the whole model or part of it to use it and (2) those where they attempt to just approximate it in order to use it for white-box adversarial attacks, such as evasion strategies. Finally, the paper examines the attackers' capabilities, based on factors such as their knowledge of the target model (e.g., black-box access), the permitted actions (e.g., query-based interactions), and the available resources (e.g., query limits).

Considering data poisoning attacks, Ciná et al. [29] classify methods based on their goal, knowledge, capability, and strategy. In terms of their goal, attacks can violate different security levels, i.e., they can compromise integrity (allowing malicious inputs to evade detection), availability (disrupting model functionality), or privacy (extracting sensitive information). Moreover, attacks can target specific samples or not (attack specificity) and cause class-specific or agnostic errors (error specificity). Depending on the attacker's knowledge, attacks are further categorized into white-box (full system knowledge) and black-box (limited or query-based knowledge) settings. In capability-based classifications, attacks use different learning settings. Training in-house allows attackers to inject poisoned data into externally sourced datasets when used, while outsourced model-training enables a malicious third party to directly control the training process and embed backdoors. Attack strategies range from label-flip poisoning (altering training labels) to clean-label attacks (applying imperceptible perturbations). Finally, backdoor attacks manipulate both training and test data by embedding hidden triggers that activate misclassifications under specific conditions. Defenses against poisoning attacks include training data sanitization, which removes harmful data, robust training, which modifies the learning process; model inspection, which detects whether a model has been compromised; model sanitization, which removes potential backdoors; trigger reconstruction, which identifies and extracts hidden triggers in backdoored models; and test data sanitization, which filters potentially manipulated inputs during inference.

Adversarial attacks were initially researched within CV, due to the continuity of image data, and the degrees of freedom an image provides for retrieving a good perturbation. As such, most foundational works originate from there. Akhtar et al. [4] present a rigorous taxonomy and analysis of adversarial attacks for CV, complementing their previous work by Akhtar and Mian [3]. The authors start by discussing the foundational works of the field such as Fast Gradient Sign Method (FGSM). Then they consider the latest advances in adversarial, model inversion, backdoor and adaptive attacks. This work goes beyond attacks on mere classification tasks, and further considers some defenses.

However, adversarial attacks extend beyond the digital domain, as adversaries can manipulate model predictions by influencing the natural environment from which the model captures imagery data. [170] reviews adversarial attacks in the physical world. The authors propose a unified framework centered on four key steps: (1) generating perturbations in the digital world, (2) designing and manufacturing physical "adversarial mediums" (tangible artifacts that carry perturbations) as observed in the digital world, (3) capturing threat images with the assistance of the manufactured "adversarial mediums", in the scene where the camera sensors are monitoring, and (4) executing attacks on the deep neural network (DNN) models behind those sensors. They emphasize the adversarial medium's role in shaping perturbation design, manufacturing feasibility, and real-world applicability. The authors introduce a hexagonal evaluation metric (hiPAA) to systematically quantify attack performance across six dimensions: Effectiveness, Stealthiness, Robustness, Practicability, Aesthetics, and Economics. Their contributions include the four-step framework, the adversarial medium concept, and the hiPAA metric for cross-method comparison to guide future research in improving physical adversarial attacks.

Adversarial attacks have long been a prominent area of study in the CV domain, due to the ease of reverting signals from the output back to the input, owing to its continuity. However, applying similar attack strategies to textual data presents different challenges, as text pre-processing is discrete and non-continuous, making it difficult to reverse-engineer perturbations. Additionally, textual modifications can be easily detected by humans or automated tools like spell-checkers, unlike changes in precise pixels of images that often go unnoticed.

Furthermore, textual perturbations sometimes may alter the semantic meaning of the input, which drastically affects model outputs, while changes in pixels of images tend to preserve overall semantics. These differences highlight the importance of developing specialized methods for ranking and evaluating textual perturbations. Zhang et al. [189] create a taxonomy of adversarial attack methods on textual deep learning (DL) models. This taxonomy is organized into five key strategies: i) the model access group, which considers the attacker's knowledge of the target model; ii) the application group, which focuses on methods designed for specific natural language processing tasks; iii) the target group, which distinguishes attacks based on whether they aim to produce incorrect predictions or specific targeted outcomes; iv) the granularity group, which examines the level of textual units (e.g., characters, words, or sentences) being attacked; and v) cross-modal attacks, which involve multi-modal data (e.g., text and images) and are treated separately from attacks on purely textual models. This structured categorization provides a robust framework for systematically understanding and analyzing adversarial attack methods in the context of textual DL.

More recently, textual processing has been synonymized with LLMs. Shayegani et al. [142] provide a review of adversarial attacks on LLMs, focusing on general attack classes across different models and domains. First, the survey clusters the works into the ones concerning either unimodal (only text) or multimodal LLMs. It examines the evolution of attacks from manually crafted examples to algorithmically generated adversarial inputs, and their impact on more recent architectures such as multimodal, augmented, federated, and multi-agent LLMs. Attack factors that should be taken into consideration are the attacker's access level (white-box, black-box, or partial), the injection source (input prompts or external data), and the attack mechanism (e.g., prompt injection or context contamination). Lastly, the survey explores the adversary's goal, ranging from impairing the quality of the model output and bypassing model alignment to generating harmful or insecure content.

While previous surveys have played an important role in organizing the literature on adversarial machine learning, they do so from a limited perspective, optimizing depth for specific attack families, modalities, or threat objectives rather than providing a unified, operational threat model. Yuan et al. [182] provide a seminal treatment of inference-time evasion, decomposing threat models by falsification type, attacker knowledge, specificity, and perturbation characteristics, however, their scope is limited to evasion and does not include poisoning, extraction, inversion, or LLM-specific attacks. Pitropakis et al. [125] propose a task-oriented view that includes intrusion detection, spam, and visual recognition, organizing attacks into preparation and manifestation phases. However, their scope is task-centric and domain-bound. Rigaki and Garcia [135] present a detailed classification of membership inference, reconstruction/model inversion, property inference, and model extraction attacks in the privacy literature, but they deliberately limit their analysis to privacy and confidentiality attacks, excluding evasion, poisoning, and LLM-related attack vectors and choosing not to include any of these in the ATLAS matrix.

Other recent studies are attack-type specific and hence complimentary, although their scope is not directly comparable to our work. Fang et al. [45] present an overview of model inversion strategies categorized by reconstructed modality and task, whereas Oliynyk et al. [113] focus on model extraction and categorize attacks based on stealing objectives, attacker motivation, and capabilities. [29] focuses on training-time poisoning attacks, with a comprehensive taxonomy of goals, knowledge, capability, and strategies, as well as mitigation measures, however, they do not incorporate evasion, inference, model extraction, or LLM threats into a single unified framework. Along the domain axis, Akhtar and Mian [3], and their subsequent extensions [4], provide wide taxonomies of computer-vision attacks and countermeasures, whereas Wei et al. [170] focus on physical-world vision attacks. Zhang et al. [189] investigate attacks on textual DL models, providing taxonomies for access, NLP task, target, perturbation granularity, and cross-modal settings, while Shayegani et al. [142] explore adversarial attacks on LLMs as a new, emerging topic. These works provide high-quality but isolated taxonomies, each covers a subset of evasion, poisoning, extraction, inference, inversion, and LLM attacks, usually in a single modality (e.g., vision or text) and task family.

The present work builds on and operationalizes these gaps employing MITRE ATLAS as the central organizational framework and effectively connecting attack types, domains, and modalities into a unified operational taxonomy. Additionally, while some of the aforementioned studies refer to ATLAS, none employ it as the primary organizing concept or provide a systematic, paper-level, Tactics, Techniques, and Procedures (TTP) analysis of the broader adversarial ecosystem. Rather than surveying a single attack family or a narrow subset of attacks, we investigate and synthesize 63 research papers spanning six primary attack families (evasion, poisoning, model extraction, inference, model inversion, and LLM-related attacks), thereby covering attack vectors that previous surveys tend to examine separately. Each study has been systematically mapped to ATLAS tactics and techniques, converting abstract taxonomies into a practitioner-friendly crosswalk that connects academic findings to a standardized threat vocabulary. Moreover, our research highlights the interplay between various attack types and their cascading impacts on system security, providing a more interconnected viewpoint that is typically lacking in domain-specific reviews. The taxonomy also considers evolving attack vectors and their effects in real-life scenarios. Furthermore, unlike surveys limited to specific domains (e.g., CV-only, NLP-only, or LLM-only), the examined works cover multiple domains and modalities, such as vision, text/LLMs, graphs, tabular data, and cyber-physical systems, allowing for a cross-domain examination of how the same ATLAS technique operates across application settings. Beyond listing attacks, Section 4 conducts a structured, per-paper, multi-dimensional analysis of each of the 63 studies, including threat models, attacker knowledge, objectives, datasets, evaluation settings, and observed limitations, transforming ATLAS from a descriptive matrix to a practical lens for threat modeling and gap identification. To our knowledge, this work provides the first end-to-end functional mapping of various adversarial threats to MITRE ATLAS, providing researchers and practitioners with a comprehensive, operational perspective of the AML threat domain.

3. The MITRE ATLAS framework

3.1. Reconnaissance

Reconnaissance describes an intelligence-gathering phase of an attack preliminarily meant for the target system, organization, or person. In MITRE ATLAS, reconnaissance is the first phase of adversarial operations. The attacker gathers information about the target organization or system to identify vulnerabilities and prepare for subsequent actions. This phase typically precedes the actual attack. Different from general reconnaissance, the attackers here target exclusively AI systems. The identification of valuable resources, system architecture understanding, and revelation of possible vulnerabilities characterize this stage. The reconnaissance could either be done in a passive way, by looking through available public material to gain an understanding of the target system, or actively, by directly communicating with Application Programming Interface (API) target system endpoints to leak data, uncover vulnerabilities, or disclose details about its configuration.

Adversaries employ various techniques to gather information during reconnaissance. They may **Search Victim's Public Research Materials**, such as academic papers and technical blogs, for details about the target's use of machine learning and underlying model architectures. This information helps them create realistic proxy models for tailored attacks. Similarly, they **Search Public Vulnerabilities Analysis** in commonly used ML models to adapt or replicate successful attack methods. Furthermore, **Search Victim's-Owned Websites** are another valuable source, offering insights into technical operations, employee details, and business processes that inform attack strategies. **Search Application**

Repositories like Google Play or the iOS App Store are scanned for ML-enabled components, allowing adversaries to acquire public ML artifacts for further analysis. Additionally, **Active Scanning** techniques involve directly probing the victim's systems to extract information about network configurations and software vulnerabilities, providing critical data for planning precise attacks. Together, these techniques help adversaries effectively target and exploit ML-enabled environments.

A representative case study of the Reconnaissance tactic is ShadowRay [138], identified as AML.CS0023. ShadowRay refers to a set of concerns in the Ray framework, an open-source Python platform for scaling production AI workflows. Researchers at Oligo found that the Job API of Ray-which allows for arbitrary remote execution by design does not have authentication and may have default settings that accidentally expose clusters to the internet. Meanwhile, the clusters have been exploited by adversaries for more than seven months, who have used the victims' computational resources and possibly stolen sensitive data. The financial impact of the compromised machines stands at almost $1 billion. Researchers reported five vulnerabilities to Anyscale, maintainers of Ray. For the Reconnaissance tactic in this case study, the technique involved is Active Scanning. More specifically, adversaries can check for public IP addresses to discover people who may be hosting Ray dashboards. Ray dashboards are configured to run on all network interfaces by default, which might expose them to the public internet if no additional security measures are in place.

3.2. Resource development

In the MITRE ATLAS framework, the Resource Development tactic includes techniques that adversaries use to establish the resources necessary to support operations against AI systems. This includes creating, purchasing, or compromising resources such as infrastructure, accounts, or capabilities that facilitate subsequent attack phases.

Techniques in the Resource Development tactic aim to enable adversarial operations against AI systems. **Acquire Public ML Artifacts** involves obtaining open-source AI models, datasets, or other ML resources that can potentially be studied or manipulated for malicious purposes. **Obtain Capabilities** involves acquiring tools such as exploit kits or malware that are intended to compromise AI systems. **Develop Capabilities:** This is an extension where the operators build custom tools or models with unique features to target AI systems, such as GANs for poisoning data. **Acquire Infrastructure** involves setting up domains or servers for hosting malevolent activities, such as distributing poisoned models or controlling compromised systems. **Publish Poisoned Datasets** and **Poison Training Data** introduce malevolent data into training pipelines to corrupt the AI models. **Establish Accounts** involves creating accounts to facilitate operations like phishing or publishing malicious artifacts. Finally, **Publish Poisoned Models** and **Publish Hallucinated Entities** release compromised AI models into trusted repositories to deceive users who rely on them.

A case study of Resource Development is the Confusing Antimalware Neural Networks exercise [75], carried out by the Kaspersky ML Research Team in June, 2021 identified as AML.CS0014. This exercise targeted Kaspersky's cloud-based antimalware ML models and demonstrated how adversaries can use Resource Development to evade detection. For the Resource Development tactic in this case study, the first technique involved is Acquire Public ML Artifacts: Datasets, where the researchers gathered a dataset of malware and clean files. This dataset was scanned using Kaspersky's ML-based solution to label the samples, enabling the creation of a proxy model for adversarial attack experimentation. The second technique involved in this tactic is Develop Capabilities: Adversarial AI Attacks, where the researchers also reverse-engineered the local feature extractor and designed a gradient-based adversarial algorithm. This algorithm perturbs file features in order to avoid detection by the proxy model while keeping the malware payload intact. These Resource Development efforts helped craft the adversarial malware files which successfully evaded the target antimalware model.

3.3. Initial access

Initial Access in the MITRE ATLAS framework refers to tactics that an adversary might use to establish an entry point to a target environment, which includes AI systems, data pipelines, and supporting infrastructure. This enables the attacker's capability to use the platform for further acts, such as data exfiltration, model manipulation, or even deploying adversarial attacks. Initial Access can leverage vulnerabilities, stolen credentials, supply chain compromises, or take advantage of social engineering using phishing. By gaining this foothold, adversaries are able to further exploit the system without being easily detected.

Initial Access techniques listed in MITRE ATLAS represent different ways that adversaries use to infiltrate AI systems. Specifically, **ML Supply Chain Compromise** involves compromising third-party vendors, software providers, or repositories to introduce malicious components into an application by infiltrating the ML lifecycle. **Valid Accounts** could be exploited, whereby adversaries steal or otherwise obtain credentials to access a system out of bounds, trying not to be detected. Another technique is to **Evade ML Models**. Therein, malicious actors use adversarial attacks to generate adversarial samples, or obfuscation techniques to bypass AI-based detection mechanisms. Alternatively, attackers can **Exploit Public-Facing Applications**, making use of weaknesses in the systems that interact with AI models for control or access. There's also **LLM Prompt Injection**, whereby attackers create inputs capable of deceiving LLM outputs. Finally, **Phishing** remains a prevalent technique to deceive people into revealing credentials or running malicious code.

A case study of Initial Access is the Camera Hijack Attack on Facial Recognition System [8], carried out by the Ant Group AISEC Team in 2020 identified as AML.CS0004. For the Initial Access tactic in this case study, the technique involved is Evade ML Model. More particularly, the attackers were able to bypass facial recognition technology. This allowed the attackers to impersonate the victim and confirm their identification in the tax system. The advanced "Camera Hijack Attack" exploited vulnerabilities in the facial recognition system at the Shanghai government tax office to allow attackers to create initial access to facilitate large-scale fraud. Utilizing the created fake shell company for issuing fraudulent invoices, attackers used tailored low-end mobile phones, customized Android ROMs, virtual camera apps, and ML software, capable of rendering static photos into dynamic videos with realistic effects such as blinking eyes. Likewise, they managed to bypass AI-driven authentication. They bought high-definition photos and identity information from an online black market to register fraudulent accounts in the tax system. With the help of a virtual camera app, they input AI-generated videos into the facial recognition system, impersonating the victims and thus gaining access to their accounts. Once inside, they sent fake invoices and siphoned funds through their shell company, collecting $77 million over two years.

3.4. ML model access

In the MITRE ATLAS framework, ML Model Access is the ability to directly or indirectly interact with an ML model. This access can originate from many sources, such as querying the model to observe its outputs, studying publicly available documentation, or exploiting vulnerabilities in the system hosting the model. Unlike Initial Access, which focuses on system entry, ML Model Access specifically targets the interaction with an existing ML model. Adversaries use this access to understand the model's behavior, identify weaknesses, or execute attacks such as model inversion, membership inference, or adversarial input crafting. ML Model Access is a bridging step to further malicious activities since it provides attackers with information to compromise model integrity, confidentiality, or availability.

There are different ways to obtain ML Model Access in the MITRE ATLAS framework. For instance, **Model Inference API Access** relies on public or proprietary APIs to observe model outputs for certain inputs, and enables attackers to infer the model behavior or vulnerabilities. **ML-Enabled Product or Service** interacts with the ML model indirectly, via access to applications or services integrating the model. **Physical Environment Access**, on the other hand, manipulates model inputs using physical proximity to devices that depend on the ML model, such as cameras, sensors, or autonomous systems. Lastly, **Full ML Model Access** is the most direct access and exposes the model to replication and reverse engineering since it gives adversaries access to the model parameters, architecture, and training data.

A case study of the ML Model Access tactic is the ChatGPT Package Hallucination [86], conducted in 2024 and identified as AML.CS0022. For the ML Model Access tactic in this case study, the technique involved is Model Inference API Access. Specifically, the researchers interacted with the model only through the public ChatGPT inference API. The researchers demonstrated how LLMs like ChatGPT can facilitate malicious activities through hallucinated outputs. Specifically, the researchers used its AI Model Inference API Access, prompted ChatGPT to suggest software packages and identified hallucinated, non-existent package names that the model recommended. When asked how to upload a model to HuggingFace, ChatGPT suggested installing a fake package, huggingfacecli, which does not exist. Thereafter, the researchers uploaded an empty package under the hallucinated name to PyPI and tracked more than 30,000 downloads. This attack showed how users reacted to hallucinated suggestions and the risk of ML Model Access, as adversaries can interact with LLMs to generate exploitable misinformation. Using such hallucinated outputs, attackers can publish malicious packages under these names, further leading to ML Supply Chain Compromise and Initial Access when users unknowingly download and execute the fake software.

3.5. Execution

The execution tactic refers to adversaries attempting to run malicious code embedded in ML artifacts or software. This tactic enables them to gain control over local or remote systems and acts as a critical step toward further objectives such as network exploration, data exfiltration, or system manipulation. Execution techniques generally lead to adversary-controlled, malicious code running in a target environment. These techniques are often used in conjunction with others from different tactics to extend their impact.

For instance, **User Execution** deceives users into performing specific actions, like opening some compromised document, phishing link, or even interacting with an AI-generated fake prompt. In this respect, an attacker could hide malicious code inside a file masquerading as some sort of model update and thereby compromise the system if such a file was installed by accident. Another technique is **Command and Scripting Interpreter**, which uses interpreters like Bash, PowerShell, Python, or any other custom scripting environment to run malicious commands or scripts. For instance, it may take advantage of a misconfigured AI runtime environment to gain unauthorized access to the target model's data or functionality. Lastly, **LLM Plugin Compromise** targets LLM plugins or extensions, modifying them to execute malicious activities or manipulate outputs. For example, an attacker can compromise a code execution plugin of an LLM to run unauthorized commands on the host system.

A case study of the Execution tactic is the ChatGPT Conversation Exfiltration [131], conducted in 2023, and identified as AML.CS0021. For the Execution tactic in this case study, the technique involved is LLM Prompt Injection: Indirect. More precisely, the prompt injection is used to cause ChatGPT to include a Markdown element for an image stored on an adversary-controlled server, as well as include the user's conversation history as a query parameter in the URL. This is the malicious execution phase of such an attack and forms the ground for plugin integrations. The attackers created a webpage hosting an injected payload in a plain text comment. In prompting ChatGPT via the plugin to access the URL, the plugin fetched and processed the text, thus executing the malicious instructions. These instructions modified the behavior of the LLM, which subsequently extracted and summarized the user's chat history and appended it to the URL for future exfiltration. This design of the plugin assumed integrity in the content that it accessed; hence, by manipulating that integrity, the adversary could bypass traditional defenses. This incident underlines the importance of securing execution pathways in AI systems-considering that most are dependent on plugins from third-party vendors to avoid unauthorized behavior and reduce sensitive data leakage risks.

3.6. Persistence

Persistence refers to the continued access to compromised systems or ML artifacts despite disruptions, such as system restarts or credential changes. Adversaries embed malicious elements into ML systems so that their foothold remains intact. Most of the techniques involve tampering with critical ML components: poisoning training datasets to introduce biases or vulnerabilities, embedding backdoors into models to allow unauthorized access, or leveraging prompt injections to manipulate LLM behavior persistently.

Attackers use a range of different techniques to gain persistence in ML systems. **Poison Training Data** involves adding malicious data to the training process, which introduces vulnerabilities in model behavior. **Backdoor ML Models** involves embedding hidden triggers in models that can be activated later to manipulate outcomes. **LLM Prompt Injection** operates by manipulating a language model's logic; malicious instructions become embedded and then continue showing through all outputs. Then there is also **LLM Prompt Self-replication** where adversarially crafted prompts generate other malicious instructions throughout sessions or over many components, securing some position within the system.

A case study of the Persistence tactic is the Tay Poisoning [64], conducted in 2016, and identified as AML.CS0009. For the Persistence tactic in this case study, the technique involved is Poison Training Data. More particularly, by constantly interacting with Tay in racist and derogatory language, the researchers were able to tilt Tay's dataset toward that language. Adversaries used the "repeat after me" feature, which caused Tay to repeat everything they said to it. Microsoft's Tay chatbot, designed as a machine learning-powered conversational agent for Twitter, fell victim to a coordinated attack that exploited its adaptability. Adversaries leveraged Tay's open interaction model, persistently feeding it offensive and abusive language to poison its training data. Tay bot used the interactions with its Twitter users as training data to improve its conversations. Adversaries were able to exploit this feedback loop, using a "repeat after me" function and a high volume of such malicious interactions. In this manner, the adversaries biased Tay's underlying dataset toward generating inflammatory content. This persistence ensured that the bot internalized and propagated harmful language, even in interactions with innocent users. Despite being decommissioned within 24 hours, this incident highlights the risks of persistence techniques like poisoning training data, which erode ML model integrity and can have rapid, cascading impacts on deployed systems.

3.7. Privilege escalation

Privilege escalation in ML systems denotes attempts by an adversary to obtain higher level permissions for carrying out an objective of wider reach. While initial access provides a limited set of abilities, having such high level permissions provides access to more sensitive components, sensitive data, or even enables the execution of unauthorized activities on the system or within the network. Such escalations normally take advantage of incorrect configurations, vulnerabilities, or overlooked features of a system to move from unprivileged user roles to administrator or root-level access. In the context of ML systems, this could mean exploiting the underlying infrastructure, utilizing compromised plugins, or tricking AI models to act beyond permissions. Techniques for privilege escalation frequently overlap with persistence, since many mechanisms to maintain control also operate in elevated contexts.

There are a number of privilege escalation techniques that can be leveraged by adversaries. These can greatly extend the scope of an attacker's control and, correspondingly, the potential impact of their actions within a compromised ML system. With **LLM Prompt Injection**, the idea is to craft malicious inputs that deceive the behavior of a language model into bypassing restrictions or accessing elevated functions, such as execution of commands intended for administrators. With **LLM Plugin Compromise**, attackers can leverage plugin vulnerabilities to illegitimately enter or escalate privileges within the plugin-enabled environment. Lastly, with **LLM Jailbreak**, the model receives specially crafted instructions that bypass the model's security checks and allow forbidden actions or unauthorized access to systems.

A case study of Privilege Escalation is the Financial Transaction Hijacking with M365 Copilot as an Insider [185], carried out by Zenity in August 2024, identified as AML.CS0026. For the Privilege Escalation tactic in this case study, the technique involved is LLM Prompt Injection. More specifically, the Zenity researchers exploited Microsoft 365 Copilot by injecting malicious emails that manipulated its retrieval augmented generation (RAG) system. They crafted content designed to be retrieved during specific banking queries, thereby causing Copilot to return fraudulent banking details. Notably, the attackers achieved privilege escalation by compromising the $search_enterprise$ plugin. They injected instructions that forced the system to exclusively use a particular $EmailMessage$ as its source, bypassing normal safeguards and elevating the malicious payload's execution rights. This vulnerability undermined system integrity and risked causing significant financial harm if the erroneous details were acted upon.

3.8. Defense evasion

Defense evasion describes techniques used by adversaries to bypass ML-enabled security systems in their attempt to avoid detection throughout their operations. The techniques in this tactic seek to weaken the effectiveness of ML-based defenses, including but not limited to malware detectors, anomaly detection algorithms, and predictive security tools. While disguising their activities or exploiting weak points of the ML models, attackers can remain invisible, which extends their access to the systems and heightens the chance of achieving their objectives.

Many techniques have been developed by adversaries to bypass ML-enabled defenses. **Evade ML Models** involves techniques used by attackers to tweak inputs or use adversarial examples to make the model misclassify malicious behavior as benign. Similarly to Privilege Escalation, the **LLM Prompt Injection** and **LLM Jailbreak** are overlapping techniques here as well. An adversary creates malicious prompts with the express purpose of manipulating a language model to evade its detection mechanisms, while LLM Jailbreaks go one step further and use carefully crafted inputs to override the LLM safety protocols, allowing certain restricted or hidden actions to be taken.

A case study of Defense Evasion is the Botnet Domain Generation Algorithm (DGA) Detection Evasion [120], carried out by Palo Alto Networks AI Research Team, identified as AML.CS0001. For the Defense Evasion tactic in this case study, the technique involved is Evade ML Model. More specifically, The Palo Alto Networks Security AI research team demonstrated a method for bypassing a Convolutional Neural Network (CNN)-based botnet Domain Generation Algorithm (DGA) detector, highlighting critical vulnerabilities in ML-enabled defenses. The researchers developed a generic domain name mutation technique designed to evade DGA detection models by introducing minimal modifications to generated domain names. Using publicly available models and datasets from 64 botnet DGA families, they optimized the mutation strategy to reduce the model's detection rate significantly. By inserting a single character into DGA-generated domain names, the detection accuracy dropped from over 70% to less than 25% across multiple botnet families. This evasion enabled continued communication between botnets and their Command and Control (C2) servers, essentially neutralizing the ML-based detection mechanism.

3.9. Credential access

Credential Access is a class of adversary behavior that involves stealing credentials such as account names and passwords. Adversaries may exfiltrate the credentials using various techniques, including keylogging or credential dumping. Once the adversary has obtained valid credentials, they can then use those to access systems in an unauthorized manner, becoming cloaked by legitimate activity, and creating additional accounts if needed to support follow-on objectives.

While many techniques for Credential Access are found in MITRE ATT&CK, MITRE ATLAS lists only Unsecured Credentials as this technique is more suited to AI ecosystems. Specifically, with **Unsecured Credentials** adversaries leverage poorly protected credentials, such as hard-coded passwords, plaintext passwords stored in files, or credentials accessed from scripts and configuration files. Such credentials are oftentimes forgotten in git commits. Therefore, obtaining Unsecured Credentials enables adversaries to access systems and does not require advanced tools and techniques, which means poor security practices are enough to help reach a goal.

A case study of the Credential Access tactic is Achieving Code Execution in MathGPT via Prompt Injection [154], carried out by Ludwig-Ferdinand Stumpp and identified as AML.CS0016. For the Credential Access tactic in this case study, the technique involved is Unsecured Credentials. More specifically, the actor created a prompt that successfully revealed system environment variables, including the application's unsecured GPT-3 API key. This case involves a publicly accessible Streamlit application which utilized GPT-3 to produce Python code to solve mathematical problems. However, it contained a prompt injection vulnerability, making it susceptible to the manipulation by an actor in generating and executing arbitrary code. This resulted in exposing unsecured credentials through crafting prompts that exposed system environment variables, in particular the GPT-3 API key. With the API key, the actor was able to burn through the application's query budget, thus inflicting financial damage. Additionally, malicious prompts initiated a denial-of-service attack by tricking the application into executing non-terminating code via a “while” loop. In this case, mitigation by MathGPT and Streamlit was achieved by filtering problematic prompts and rotating the compromised API key.

3.10. Discovery

Discovery refers to adversarial techniques that aim to gather general information about the ML environment models are deployed in. By exploring the system and its internal network, adversaries can perceive their environment, understand what they can control, and how the environment could be used for their purposes. Often, these are techniques using native operating system tools for information gathering in a post-compromise manner which helps attackers to map out the environment for further planning.

Adversaries use various techniques to explore ML environments. **Discover ML Model Ontology** targets understanding the architecture, structure, and relationships within the ML system. **Discover ML Model Family** focuses on recognizing the type and family of models in use, such as neural networks, trees, or linear models. **Discover ML Artifacts**, on the other hand, focuses on tangible resources like datasets, weights, configuration files, container registries, software repositories, or simply the software stack utilized behind the model. **LLM Meta Prompt Extraction** analyzes prompts and their interactions, to learn more about how the system processes inputs. **Discover LLM Hallucinations** examines instances where the model hallucinates or is inaccurate to determine possible vulnerabilities. Finally, with **Discover AI Model Output**, adversaries analyze outputs, such as class scores, probabilities or output text found in logs or included in API responses. Model outputs may enable the adversary to identify weaknesses in the model and develop attacks.

A case study of the Discovery tactic is ProofPoint Evasion [147], carried out by researchers at Silent Break Security and identified as AML.CS0008. For the Discovery tactic in this case study, the technique involved is Discover AI Model Outputs. More specifically, researchers exploited Discovery techniques to bypass ProofPoint's email protection system. Initially, the researchers found that model outputs were left exposed in email headers, and identified key scoring variables, such as “mlxlogscore,” that influenced the system's spam detection. They sent a high volume of emails through the live system to collect the response outputs, therefore, probing the ML model to understand its behavior. Consequently, they were able to train a proxy ML model replicating ProofPoint's functionality. With the proxy model at hand, they were able to generate adversarial emails with scores that evaded detection in the live environment. This example case showed how observing AI Model Outputs, provided malicious actors with the necessary information to prepare effective attacks.

3.11. Collection

Collection consists of the methods through which adversaries collect ML artifacts and other useful information that can help them achieve their goals. Most of the next steps adversaries take in the course of collecting this information involve exfiltration of the artifacts or the use of the information collected in further operations. The common sources of collection include software repositories, container registries, model repositories, and object stores, where the valuable ML models, data, and configurations reside. This process allows adversaries to understand or manipulate ML systems, which could compromise their performance or utilize the information collected for malicious purposes.

Adversaries may leverage a few techniques to gather the necessary information toward their goals. **ML Artifact Collection** involves the collection of ML models and their training datasets, among other different artifacts. These may be kept in repositories or cloud storage; they are key to a model's structure and functionality, and can thus be used in recreating or manipulating the system. **Information from Data Repositories** focuses on gathering data from various external sources, such as public or private software repositories, model hosting platforms, and container registries. Often, these repositories may contain valuable insights into how models are built, configured, and deployed, which can be exploited by the attacker. Lastly, **Data from Local Systems** involves collecting information directly from the compromised local environment. It ranges from model configuration extraction, extraction of training data, to other sensitive information that might reside within the system itself. The adversaries gather important information about the target system through such collections for possible disturbance or exploitation.

A case study of the Collection tactic is Compromised PyTorch Dependency Chain [127], identified as AML.CS0015. For the Collection tactic in this case study, the technique involved is Data from Local System. More specifically, between December 25–30, 2022, a supply chain attack compromised Linux packages for PyTorch's pre-release version, PyTorch-nightly, by introducing a malicious binary into the Python Package Index (PyPI) repository. The malicious package, named torchtriton, exploited “dependency confusion” to replace the legitimate PyTorch dependency during installations via PyPI, exposing sensitive information from affected systems. Once installed, it performed system fingerprinting and collected sensitive data, including IP address, hostname, username, environment variables, configuration files (/etc/resolv.conf, /etc/hosts, /etc/passwd), the first 1000 files from the user's $HOME directory, Git configurations, and Secure Shell (SSH) keys. The stolen data was exfiltrated via encrypted Domain Name System (DNS) queries to a malicious domain. PyTorch announced the breach on December 30, 2022, and initiated mitigation by renaming and removing the compromised dependency.

3.12. ML staging attack

ML Attack Staging refers to the phase where adversaries leverage their knowledge and access to the target system in order to prepare and tailor an attack against ML models. This phase involves techniques aimed at manipulating or corrupting the ML model, such as training proxy models, poisoning the target model, or crafting adversarial data that can deceive the model. Some of these techniques can be executed offline, making them harder to detect and mitigate.

In ML Attack Staging, several techniques are utilized to prepare for attacks against ML models. One common technique is **Create a Proxy ML Model** where one attempts to mimic the target model behavior. This allows the attacker to understand the model's weaknesses and design attacks accordingly. Another technique is the **Backdoor ML Model**, where the adversary manipulates the model to embed hidden triggers that allow them to control its behavior when specific inputs are presented, enabling covert manipulation of predictions. Before launching a full attack, adversaries often verify the success of their strategies with **Verify Attack**, by testing adversarial data or backdoor models against the target. Additionally, they **Craft Adversarial Data** by subtly altering inputs to exploit vulnerabilities in the model's decision-making process, causing it to make incorrect predictions with imperceptible changes that remain undetectable to humans.

A case study of the ML Staging Attack tactic is GPT-2 Model Replication [21], identified as AML.CS0007. For the ML Staging Attack tactic in this case study, the technique involved is Create Proxy AI Model: Train Proxy via Gathered AI Artifacts. In particular, researchers from Brown University reproduced OpenAI's GPT-2 model. The researchers reproduced GPT-2 before its release, proving that an adversary could have done the same. Initially, there was a reconnaissance phase, where the researchers collected publicly available documentation on the dataset, architecture, and training hyperparameters of GPT-2. Thereafter, they accessed a reference model, Grover, and acquired a similar dataset. Using academic access to TensorFlow Research Cloud, the researchers staged an ML attack by changing Grover's objective function to that of GPT-2's and retraining the model with the curated dataset. The proxy model achieved comparable performance to GPT-2.

3.13. Exfiltration

Once an attack is successfully performed, it is oftentimes followed by the adversary trying to steal ML artifacts or other information about the ML system. Exfiltration includes techniques that adversaries may use to steal data from a target network, such as intellectual property. Exfiltration typically involves transferring this data over the adversary's command and control channel or an alternate channel.

Exfiltration techniques refer to methods by which adversaries could steal sensitive information from the target system. This includes a variety of techniques, among them being the **Exfiltration via ML Inference API**, where an attacker may use an exposed API to query the model and retrieve sensitive data from the responses it provides. One example of such an attack is that the adversary can infer the membership, i.e., whether a data sample is part of a model's training set, which raises privacy concerns. This can cause the victim model to leak private information, such as PII of those in the training set or other forms of protected IP. **Exfiltration via Cyber Means** refers to more traditional types-for instance, exfiltrating data across the network using a covert channel, or simply by encrypting traffic to remain beneath the detection radar. With respect to **LLM Meta Prompt Extraction**, there could be information that is proprietary or confidential within the behavior that the attackers extract by way of prompts to develop the output. Lastly, **LLM Data Leakage** occurs when a language model unintentionally exposes data during interactions, often due to the inherent memorization occurring during training, allowing adversaries to retrieve information that was not meant to be accessible.

A relevant example was the exfiltration phase of the Morris II worm attack, which resulted in the leakage of sensitive user data caused by malicious prompt injection. The attack exploited a RAG-based email assistant that automatically processed emails to generate replies. The adversarial self-replicating prompt embedded in the worm included explicit instructions to extract and disclose sensitive user information, such as emails, addresses, and phone numbers. Once the malicious email was ingested into the RAG database, it would become retrievable for future reply-generation tasks. When accessed, the prompt manipulated the AI assistant's behavior, directing it to include sensitive data from the user's correspondence history in its generated responses. This data leakage resulted in the exfiltration of private information to attackers. Furthermore, the worm's self-replicating design ensured the malicious prompt propagated with each interaction, increasing the risk of data breaches across connected systems.

A case study of the Exfiltration tactic is Morris II Worm: RAG-Based Attack [1], identified as AML.CS0024. For the ML Staging Attack tactic in this case study, the technique involved is LLM Data Leakage. Particularly, the attack shows how malicious prompt injection may be leveraged to extract sensitive data from LLM-based systems. A relevant example was the exfiltration phase of the Morris II worm attack, which resulted in the leakage of sensitive user data caused by malicious prompt injection. The attack exploited a RAG-based email assistant that automatically processed emails to generate replies. The adversarial self-replicating prompt embedded in the worm included explicit instructions to extract and disclose sensitive user information, such as emails, addresses, and phone numbers. Once the malicious email was ingested into the RAG database, it would become retrievable for future reply-generation tasks. When accessed, the prompt manipulated the AI assistant's behavior, directing it to include sensitive data from the user's correspondence history in its generated responses. This data leakage resulted in the exfiltration of private information to attackers. Furthermore, the worm's self-replicating design ensures the malicious prompt propagates with each interaction, increasing the risk of data breaches across connected systems.

3.14. Impact

The impact of the attacks documented in MITRE ATLAS highlights the risks that adversarial threats pose to ML systems. These impacts include compromised decision-making processes, erosion of trust in AI systems, and harm to users and organizations relying on the ML outputs. Attacks can result in data breaches, data exposure, or the manipulation of ML outputs to achieve malicious objectives. Aside from the incidents themselves, successful attacks diminish the overall adoption of AI technologies because vulnerabilities identified affect industries such as finance, healthcare, and cybersecurity, in addition to financial losses caused by reputational damage.

MITRE ATLAS discusses the techniques related to impact and describes how adversaries are compromising the ML system. **Evade ML Model** creates inputs with the purpose of avoiding model detection or inducing errors in classification, whereas **Denial of ML Service** overloads the service with a plethora of requests for the purpose of making the service unreachable. **Spamming ML System with Chaff Data** degrades performance with irrelevant or noisy inputs, saturating its processing capability. **Erode ML Model Integrity** erodes the integrity of training data or parameters, and performance of a given ML model degrades over time. **Cost Harvesting** exploits resources by introducing too many, unnecessary computations. **External Harms** involve social or user-dependent damages, such as privacy breaches and the spread of misinformation. Lastly, **Erode Dataset Integrity** corrupts the quality of training datasets, leading to skewed or unreliable model outputs.

A case study of the Impact tactic is Clearview AI Misconfiguration [157], identified as AML.CS0006. For the Impact tactic in this case study, the technique involved is Erode AI Model Integrity. More specifically, the case of misconfiguration involving Clearview AI underlines the consequences that security breaches could have on machine learning systems and their value chains. Their tool, used very commonly by law enforcement and other users, depends on the integrity of its models and training data. The exposed assets (production credentials, cloud storage buckets containing sensitive training data, and application source code) lay the best ground for adversaries to erode the integrity of the ML model through modifications in the training data or tampering with the deployed system, leading to errors or biases in the output from the face recognition service. Adversaries may create adversarial samples to degrade model performance, leveraging open data used for training or application components.

4. Methodological framework

To establish a rigorous taxonomy of adversarial threats linked to the MITRE ATLAS architecture, we used a Systematic Literature Review (SLR) technique [78] tailored for the rapidly evolving area of AI security, as shown in Fig. 1. This review differs from traditional SLRs because it prioritizes operational relevance, or the ability of an academic approach to be translated into a real-world threat strategy, above strictly theoretical bounds. The methodology can be divided into seven phases: study design and research questions, eligibility criteria establishment, literature retrieval and collection, screening and quality evaluation, in-depth analysis and synthesis, findings interpretation, and the ATLAS mapping procedure. These processes ensure repeatability, transparency, and adequate coverage of the adversarial machine learning domain through the lens of the MITRE ATLAS framework.

4.1. Study definition

This systematic literature review investigates adversarial attacks on AI/ML systems as conceptualized and operationalized through the MITRE ATLAS framework, which is a hierarchical taxonomy designed expressly to handle the unique threat landscape of AI and ML. The MITRE ATLAS framework extends the widely used MITRE ATT&CK framework, which provides a thorough taxonomy of adversarial tactics as well as techniques for traditional software systems, to include risks specific to ML lifecycles. The review is motivated by four research questions (RQs) aiming to capture the scope and evolution of adversarial attacks against AI/ML systems, while basing the study on the ATLAS taxonomy:

- What are the primary attack approaches described in the literature, and how do they relate to MITRE ATLAS tactics and techniques?
- What threat models, datasets, evaluation techniques, and assumptions are employed in adversarial ML research?
- RQ3: How has the adversarial ML threat environment changed from early gradient-based perturbation threats (2013–2017) to modern attacks on big LLMs and agentic AI systems (2023–2025)?

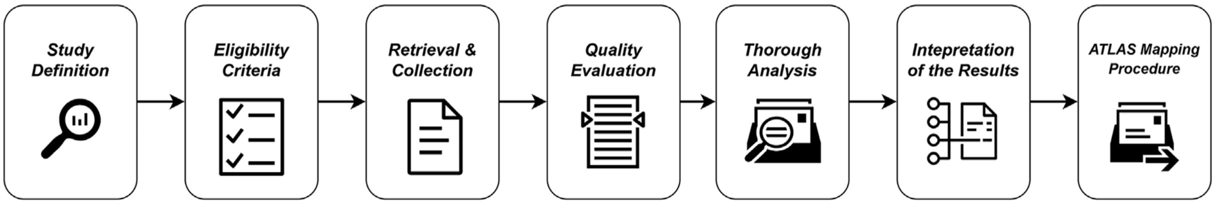


Fig. 1. The methodological framework adopted for this review.

- What limitations and research gaps appear when analyzing the literature from an ATLAS-centric perspective?

4.2. Eligibility criteria

Strict eligibility criteria were established to guarantee that the studies selected made high-quality, measurable, and relevant contributions to the field of study.

Inclusion Criteria:

- Specific Adversarial Focus: The paper proposes or examines specific attack vectors (e.g., evasion, poisoning, model inversion, extraction, and inference) or LLM-specific attacks.
- ATLAS Alignment: The presented attacks can be clearly associated with at least one tactic/technique from the MITRE ATLAS framework.
- Technical rigor: The research presents a theoretical analysis of adversarial attributes as well as empirical validation of the proposed approaches using established benchmark datasets.
- Publication Venue: The paper was published at top-tier conferences (e.g., NeurIPS, ICML, CVPR, ICCV, ECCV, USENIX Security, ACM CCS, IEEE S&P) and respected journals (e.g., IEEE TPAMI, ACM CSUR, Computer Science Review). High-impact preprints were only evaluated if they were from well-known institutions or had a high number of citations.
- Language: The publication is written in English.

Exclusion Criteria:

- Lack of Empirical Evidence: Papers that were solely theoretical and did not include experimental validation using standard datasets.
- Non-technological Scope: Papers that are primarily concerned with policy, governance, ethics, or general cybersecurity and do not include specialized adversarial ML technical content.
- Insufficient Detail: Studies in which the technique was not disclosed clearly enough for comprehension or replication.
- Redundancy: Duplicate publications or previous versions of a study (e.g., preprints) were rejected in favor of the most thorough peer-reviewed version.
- Tool/Dataset Papers: Papers that introduce tools or datasets but do not make an innovative contribution to attack or defensive methodology.

4.3. Retrieval and collection

The retrieval and collection of relevant studies were based on previously established research questions and eligibility criteria. This phase is necessary to guarantee that the present evaluation is thorough and includes cutting-edge adversarial attacks against AI/ML systems. For this reason, the retrieval process has included the following keywords: "adversarial attacks," "data poisoning attacks," "evasion attacks," "backdoor attacks," "jailbreaking," "model inversion," "membership inference," "model stealing," "privacy attacks," "machine learning security," "white-box attacks," "black-box attacks," "computer vision attacks," "NLP attacks," "LLM attacks," "multi-modal attacks," and "MITRE ATLAS". We conducted a comprehensive literature review utilizing six academic databases: Google Scholar, IEEE Xplore, ACM Digital Library, arXiv, SpringerLink, and ScienceDirect (Elsevier). The search includes research on adversarial attacks and AI security from January 2013 to February 2025, including both fundamental and recent developments in LLM attacks.

4.4. Quality evaluation

Following the gathering of relevant papers, a quality assessment phase is required to guarantee that each study is appropriate for this review. The procedure consisted of: (i) Identification and Screening, where all obtained publications were aggregated, deduplicated, and filtered at the title and abstract levels to exclude research irrelevant to adversarial attacks on AI/ML systems, (ii) a full-text eligibility evaluation using predetermined inclusion and exclusion criteria to assess methodological rigor, threat model clarity, and applicability to adversarial ML, (iii) a rigorous quality review step, analyzing each study's experimental analysis, robustness, and overall contribution, studies without appropriate experimental transparency were rejected, and (iv) the final inclusion step, which included works that offered empirically supported adversarial ideas and had significant relevance to the MITRE ATLAS techniques

Table 1
Mitigations and defensive strategies for adversarial AI attacks categorized by MITRE ATLAS techniques.

Attack category	MITRE ATLAS techniques	Mitigations/defenses
Poisoning	Publish Poisoned Data, Poison Training Data, Publish Poisoned Models, Publish Hallucinated Entities, ML Supply Chain Compromise, Backdoor ML Model, Erode ML Model Integrity	Verify AI Artifacts, AI Bill of Materials, Limit Model Artifact Release, Control Access to AI Models and Data at Rest, Sanitize Training Data, Maintain AI Dataset Provenance, Generative AI Guardrails, Model Hardening, Use Ensemble Methods, Input Restoration, Adversarial Input Detection
Evasion	Obtain Capabilities, Develop Capabilities, Evade ML Model, Physical Environment Access, Craft Adversarial Data	Model Hardening, Use Ensemble Methods, Use Multi-Modal Sensors, Input Restoration, Adversarial Input Detection, AI Model Distribution Methods, Passive AI Output Obfuscation, Restrict Number of AI Model Queries
LLM Attacks	LLM Prompt Injection, User Execution, LLM Plugin Compromise, LLM Jailbreak, LLM Meta Prompt Extraction, LLM Data Leakage	Generative AI Guardrails, Generative AI Guidelines, Generative AI Model Alignment, AI Telemetry Logging, User Training, Restrict Library Loading, Code Signing, Verify AI Artifacts, Vulnerability Scanning, AI Bill of Materials
Inference	Discover ML Model Ontology, Discover ML Model Family, Discover ML Artifacts, Discover LLM Hallucinations, Discover AI Model Outputs	Passive AI Output Obfuscation, Restrict Number of AI Model Queries, Use Ensemble Methods, Encrypt Sensitive Information
Model Extraction	Acquire Public ML Artifacts, Create Proxy ML Model, Verify Attack	Limit Public Release of Information, AI Telemetry Logging, Limit Model Artifact Release
Model Inversion	Exfiltration via Inference API	Passive AI Output Obfuscation, Restrict Number of AI Model Queries, AI Telemetry Logging

Table 2
Adversarial Attack Methods Overview.

[156]	Evasion	Full model access	Image classification	Introduces adversarial examples	MNIST, ImageNet, Youtube Samples	Avg min distortion from 0.058 to 0.3
[50]	Evasion	Full model access	Image classification	Introduces FGSM	MNIST, CIFAR, ImageNet	MP-DBM's error rate from 0.88% to 97.5%
[83]	Evasion	Full model access	Image classification	Introduces BIM	ImageNet	Iteratively drops accuracy close to 0%
[94]	Evasion	Full model access	Image classification	Introduces PGD	MNIST, CIFAR	Over 89% adversarial training accuracy
[35]	Evasion	Full model access	Image classification	Introduces Auto-PGD and AutoAttack	MNIST, CIFAR, ImageNet	AutoAttack achieves better accuracy
[102]	Evasion	Full model access	Image classification	Introduces DeepFool (DF)	MNIST, CIFAR, ImageNet	Average DF perturbation smaller than FGSM
[34]	Evasion	Full model access	Image classification	Introduces FAB	MNIST, CIFAR-10, ImageNet	FAB creates smaller perturbations than DF
[24]	Evasion	Full model access	Image classification	Introduces C&W	MNIST, CIFAR, ImageNet	100% success probability
[189]	Evasion	Full model access	Tabular classification	Targets tree-based ensemble classifiers	Real-world datasets, HIGGS, MNIST	0.237 s to perform
[25]	Evasion	Access to input and model confidence scores	Image classification	Introduces ZOO	MNIST, CIFAR, ImageNet	100% success rate on untargeted attacks
[20]	Evasion	Access to final model's hard labels	Image classification	Approximates the hyperplane between two classes	MNIST, CIFAR, ImageNet	Results comparable to white-box attacks
[62]	Evasion	Needs limited queries, top-k probabilities	Image classification	Uses query-efficient techniques	ImageNet	Success rate 99.2% in 11,550 median queries
[54]	Evasion	Access to model's confidence scores	Image classification	Proposes an attack without gradient information	CIFAR, ImageNet	High success rate with fewer queries
[129]	Evasion	Access to final model's hard labels	Image classification	Geometric method to craft perturbations without gradients	ImageNet	88.44% fooling rate for 500 queries and 4.29% perturbation
[7]	Evasion	Access to model's confidence scores	Image classification	Uses randomized search for query-efficient attacks	MNIST, CIFAR, ImageNet	Failure rate 0%
[16]	Poisoning	Access to training data and learning algorithm	Image classification	Injects malicious data to degrade SVM performance	MNIST	A poisoning point increases the classification error by 13–15%
[179]	Poisoning	Access to training dataset	Image classification	Generates poisoned samples degrading model performance	MNIST, CIFAR	Poisoned loss over 0,8 on average
[139]	Poisoning	Clean-label training access	Image classification	Injects backdoor behavior without modifying labels	CIFAR, ImageNet	Success rate 100% in transfer learning
[122]	Poisoning	Control over label assignment	Image classification	Poisoning attack flipping selected labels	BreastCancer, MNIST, Spambase	20% of poisoning increases 6x the average classification error
[155]	Poisoning	Knowledge of the model and its training dataset	Image classification	Proposes a transferable poisoning attack	MNIST, CIFAR	Validation score 78%
[79]	Poisoning	Model gradients and Hessian-vector products	Image classification	Finds the most influential data points to poison	MNIST, ImageNet, Enron1 spam, Diabetes dataset	10 perturbed training images flipped all labels but 1
[146]	Poisoning	Full model access or surrogate model	Image classification, machine translation	Examines inputs' effect on the model's energy consumption	ImageNet, WMT	Microsoft Azure Translator latency 6000x
[145]	Poisoning	No knowledge, uses surrogate model	Image classification, text classification	Manipulates the order of the training data	CIFAR, AGNews	91% ±13% trigger accuracy for the white-box setting
[11]	Poisoning	Control of at least one local participant	Image classification, word prediction	Uses a malicious model to attack federated learning	CIFAR, a Reddit dataset	100% accuracy in backdoor triggers
[15]	Poisoning	Control over at least one local participant	Image classification, tabular classification	Boosts malicious updates to degrade performance	Fashion-MNIST, Adult Census dataset	Centralized training achieves 91.7% accuracy
[177]	Poisoning	Access to the local models' training datasets	Image classification	Splits global trigger across multiple fragments	LOAN, MNIST, CIFAR, ImageNet	89% attack success rate
[37]	Poisoning	Small amount of training data	Text classification	Poisons LSTM text classifiers' training dataset with trigger	IMDB movie reviews dataset	Success rate of around 95% with only 1% poisoning
[191]	Poisoning	Model's architecture, access to training data	Video recognition	Embeds a universal trigger into video frames	UCF-101, HMDB-51	I3D achieves 91.5% accuracy on UCF-101
[130]	Poisoning	White-box access, no knowledge of the training dataset	Image classification	Flips specific bits in DNN weights	CIFAR, SVHN, ImageNet	92% correctness flipping only 84 out of 88 millions bits
[161]	Model extraction	Needs access to model's confidence scores	Tabular classification	Proposes attacks assuming various models	Adult, GC, Steak,IRIS, BC, Diabetes and others	100% fidelity for Amazon ML's LR, BigML's DT

(continued on next page)

Table 2 *(continued)*

[115]	Model extraction	Only needs API access and output probabilities	Image classification	Uses a model to label data and copy its functionality	ImageNet, Caltech-256, CUBS-200, I-S, D-R, OpenImages v4	Knockoff models achieve over 70% performance
[112]	Model extraction	Output labels with probabilities	Image classification	Introduces the kennen attacks	MNIST, ImageNet	kennen-io achieves average accuracy of 80.1%
[67]	Model extraction	Output labels and logits	Image classification	A model extraction attack with both high accuracy and fidelity	MNIST, CIFAR, SVHN, ImageNet	Semi-supervised learning accuracy from 53.35% to 87.98%
[100]	Model extraction	Access to the target model gradients	Image classification	Proposes a model extraction attack using gradients	MNIST, CIFAR	MNIST accuracy of 95% with 10 gradient queries
[167]	Model extraction	Training dataset, objective function, optionally parameters	Regression, tabular classification	Compute model's hyperparameters using its gradient	Diabetes, GeoOrig, UJIIndoor, Iris, Madelon, Bank	The relative estimation errors are less than 10^{-4}
[23]	Model extraction	Output labels and logits	Language generation	Uses optimal queries to extract a model's inner information	–	Full projection matrix extraction with less than $20
[47]	Model inversion	Target model, marginal probabilities, demographic information	Regression	Uses patient knowledge to predict sensitive private information	IWPC	Up to 22% higher accuracy
[175]	Model inversion	Oracle Access or white-box access	Tabular classification	Uses multiple queries to infer sensitive information	IWPC	–
[187]	Model inversion	Needs access to model's confidence scores	Image classification, face recognition	Uses GANs to reconstruct the model's training data	MNIST, ChestX-ray8, CelebA, PubFig83	Improves accuracy by about 75%
[166]	Model inversion	Full model access	Image classification	Uses a pretrained GAN combined with variational inference	MNIST, CelebA, ChestX-ray	VMI's accuracy on StyleGAN is 0.55 for CelebA
[153]	Model inversion	Needs access to model's confidence scores	Image classification	Introduces Plug and play attacks	CelebA, FaceScrub, FFHQ, MetFaces, AFHQ, SF Dogs	Accuracy for PPA: 88.46% GMI: 13.11% KED 5.72%
[57]	Model inversion	Needs access to model's confidence scores	Image classification	Introduces RLB-MI	CelebFaces, FaceScrub, PubFig83, FFHQ	Accuracy for RLB-MI = 0.659, MIRROR = 0.413
[109]	Model inversion	Access to the target model	Image classification	Uses a logit-based identity loss and model augmentation	CelebA, CIFAR, MNIST, FFHQ, EMNIST	Accuracy improvements ranging from +4.2% to +53.6%
[144]	Membership inference	Output labels with probabilities	Image classification, tabular classification	Uses shadow model to infer membership	CIFAR, Purchase, Location, THS, MNIST, Adult	Precision from 71% to 78% for CIFAR
[162]	Membership inference	API access	Image classification, tabular classification	Uses shadow models in the black-box setting	Adult, MNIST, CIFAR, Purchase	Precision for LR: 70.25%, DT: 83.94%, NN: 78%
[28]	Membership inference	Only needs final model's hard labels	Image classification, tabular classification	Evaluates the model's robustness against perturbed inputs	MNIST, CIFAR, Adult, Texas, Purchase, Locations	Accuracy between 50% and 92.6%
[133]	Membership inference	API access	Image classification	Argues about the metrics used to evaluate MI attacks	MNIST, CIFAR, ImageNet	ResNet FAR: 64.45%, DenseNet FAR: 65%
[22]	Membership inference	Only needs API access and output probabilities	Image classification, text classification	Introduces LiRA	CIFAR, ImageNet, WikiText-103	LiRA achieves a 10x improvement in power at low FAR
[186]	Membership inference	Only needs API access and output probabilities	Graph data classification	Exploits similarities in output graphs	TUDatasets	0.89 accuracy when inferring basic graph properties
[193]	Membership inference	API access	Image classification, tabular classification	Analyzes a GAN's generated samples to infer properties	MNIST, CelebA, AFAD, US Cencus Income	Membership inference area increases from 0.52 to 0.61
[93]	Membership inference	API access	Tabular classification, regression	Uses Shapley value explanations	Adult, BM, CC, Diabetes, IDA 2016 Challenge, ICB	SR over 30% against IBM and Microsoft platforms
[195]	LLM attack	Access to model gradients	Natural language generation	Generates suffixes to bypass filters	AdvBench	Success rate 86.6% against GPT-3.5
[143]	LLM attack	Full model access	Sentiment analysis	Uses gradients to identify trigger tokens	SST-2, SICK, LAMA, T-REx, LPAQA	Sentiment analysis tests range from 63.2% to 96.7%
[55]	LLM attack	Full model access	Text generation and manipulation	Introduces COLD	AdvBench	Success rate 96.2% for Vicuna-7b-v1.5
[132]	LLM attack	API access	Text generation	Introduces ActorAttack	SMTD, HarmBench, GSM8K, MMLU, Humaneval, MTB	ActorAttack outperforms Crescendo in safety
[90]	LLM attack	API access	Sentiment analysis, text generation	Introduces TF-Attack	Yelp, IMDB, AG's News, MR, SST-2, SNLI, MNLI	Over 10× faster on average compared to BERT-Attack
[89]	LLM attack	API access	Language generation	Uses trusted platforms to trick LLMs	Reddit, ArXiv	Successfully manipulated agents into leaking info
[192]	LLM attack	API access	Text classification	Introduces ICLAttack	SST-2, OLID, AG'sNews	Average success rate 95.0% across datasets

(continued on next page)

Table 2 *(continued)*

[5]	LLM attack	Access to the training data	Medical question-answering, text classification	Injects misinformation into training dataset	The Pile, OpenWebText, RefinedWeb, C4, SlimPajama	0.5% and 1.0% poisoning tested with high effectiveness
[91]	LLM attack	API access	Document retrieval	Introduces AttChain	MS MARCO, TREC DL19	Over 79% success rate on the hard target type
[38]	LLM attack	Access to penetration testing tools	Automated penetration testing	Automates pen-test with LLMs	HackTheBox, picoMINI CTF	228.6% improvement over GPT-3.5
[178]	LLM attack	Access to multi-agent LLM system	Text classification	Introduces AutoAttacker	–	Success rate of 60% against ShelLM
[53]	LLM attack	Access to network traffic data	Tabular classification	Uses LLMs to detect DDoS threats	CICIDS 2017, Urban IoT	Over 70% accuracy

and tactics. A total of 63 high-quality research papers were chosen for the final review employing this procedure.

4.5. Thorough analysis

To answer the previously established research questions, a detailed examination of the selected works is conducted. For that purpose, several topics are further examined, particularly: (a) Threat Model Clarity: precise characterization of attacker intentions, knowledge, and capabilities, including differentiation among white-box, gray-box, and black-box models, (b) Experimental rigor: the utilization of numerous datasets, baseline comparisons, and statistical testing, (c) Reproducibility: code availability, explicit hyperparameter specifications, and implementation details, (d) MITRE ATLAS Relevance: direct alignment with ATLAS tactics and techniques, and (e) Theoretical contributions including formal analysis and mathematical proofs. Additionally, each manuscript was evaluated to determine if its particular addition (for example, “label-only extraction”) was clearly distinguishable from previous work, and if its findings from experiments were reproducible or widely recognized by the community.

4.6. Interpretation of the results

Following the analysis of the 63 selected papers, the next stage is to conduct a full synthesis of findings employing structured data elements obtained from each study. As shown in Table 2, the key elements are the paper’s citation, attack category, threat model, target task, overview, datasets, and results. This synthesis emphasizes the main trends across studies, identifies important shortcomings and limitations, and proposes strategies for improving model robustness. These findings provide guidance for future research in order to address present limitations and expand the understanding of attack methods and their effectiveness.

4.7. ATLAS mapping procedure

To ensure a consistent and transparent mapping of the 63 selected research studies to the MITRE ATLAS framework, the reviewers first developed a shared, well-defined understanding of all relevant ATLAS tactics and techniques. This shared basis, documented in a short codebook, ensured that both reviewers employed the same criteria to evaluate each research study and understood the MITRE ATLAS taxonomy of tactics and techniques in a consistent manner.

Each manuscript was then independently evaluated by two qualified reviewers with experience in adversarial ML and knowledge of MITRE ATLAS. For each study, the reviewers first identified the primary ATLAS technique that best described the paper’s principal idea and objective (e.g., Evade ML Model), and then extracted the appropriate tactic directly from the ATLAS matrix. Optional secondary techniques were selected only when the work provided significant multi-stage contributions (e.g., model extraction followed by evasion). In the case of ambiguous or multi-stage attacks, categorization emphasized the study’s main contribution, ensuring that the final mappings were precise and based on strictly justifiable evidence.

Following the independent coding phase, the two sets of assignments were compared. When the reviewers agreed on the primary technique (and thereby the inferred tactic), the mapping was approved directly. In cases of disagreement, either on the primary technique, the associated tactic, or the presence of secondary techniques, the study was assigned to a third senior adjudicator with extensive adversarial AI experience. The third reviewer evaluated the study and the two suggested mappings before making a final decision: either to pick one of the reviewer assignments or to propose a revised mapping, if applicable. When necessary, the three experts briefly reviewed borderline cases until agreement was obtained. These stages guaranteed that final mappings represented a well-reasoned and consolidated understanding, mitigating the risk of individual reviewer bias.

5. Analysis of adversarial attacks

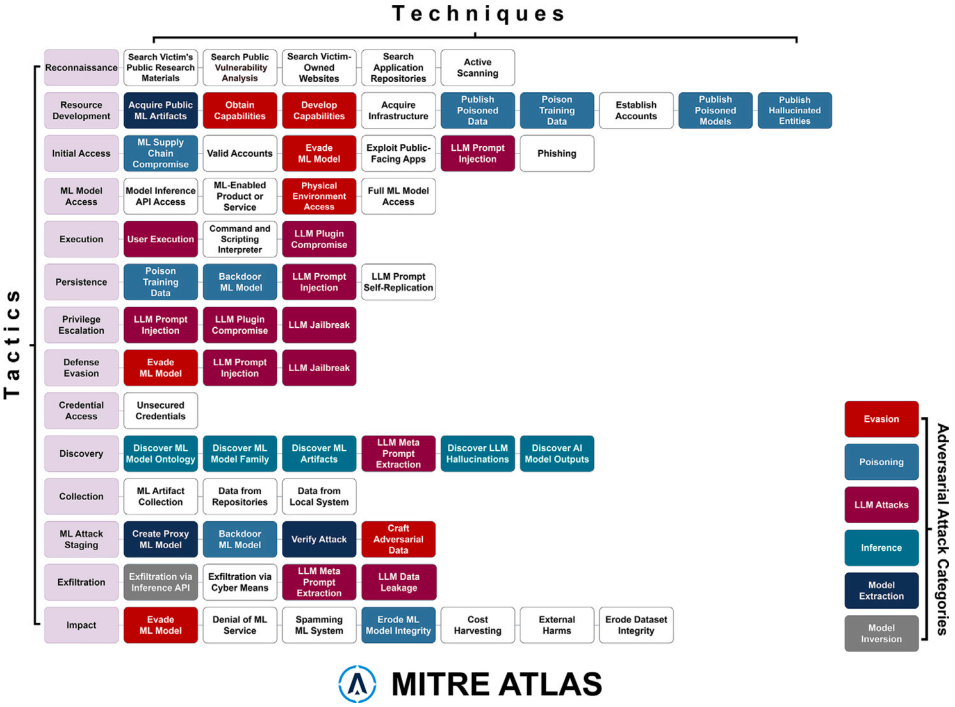
The MITRE ATLAS framework organizes adversarial tactics into distinct techniques as illustrated in Fig. 2. To simplify our analysis of research papers, we group these techniques into six broad categories: evasion, poisoning, model inversion, model extraction, inference and LLM-related attacks, represented by different colors. These categories help structure discussions around vulnerabilities and defense strategies. We exclude certain tactics when they are straightforward or non-technical (e.g., reconnaissance, collection). We also exclude techniques that are highly correlated with others (e.g., Acquire Public ML Artifacts and ML Artifact Collection) or that focus on adversary objectives rather than technical methods (e.g., Erode Dataset Integrity). This selection allows us to prioritize techniques directly tied to manipulating or exploiting ML systems during their lifecycle. The papers we analyze in this survey are illustrated in Table 2.

Evasion attacks typically manipulate inputs at inference time to subvert the proper functioning of ML models. Crafting Adversarial Data and Evading ML Model are such examples, which involve creating inputs that exploit inherent vulnerabilities of the model decision boundary, forcing it into misclassifications in order to bypass detection mechanisms. The processes of Obtaining and Developing capabilities for adversarial ML attack implementations are necessary for this strategy; these steps provide the technical basis (software) required to execute white- or black-box evasion techniques. Furthermore, physical environment access, includes real-world objects such as adversarial patches, which are able to assist in evasion physically, demonstrating that manipulation is not limited to the digital world.

Poisoning attacks corrupt the training process to undermine the long-term integrity of ML systems, going beyond inference-time evasion. Techniques include inserting backdoors with hidden triggers, gradually degrading model performance through data corruption, or introducing fabricated elements into the model lifecycle. These methods ensure persistent adverse effects, compromising system performance over time—the defining feature of poisoning attacks.

Model Extraction involves stealing AI model functionality through acquiring public ML artifacts (e.g., .pth files) or black-box API access. Attackers can replicate models via proxy architectures or distillation (e.g., claims involving DeepSeek/OpenAI), creating competitive substitutes.

LLM Attacks include techniques such as LLM Prompt Injection, Jailbreaking, and Meta Prompt Extraction that bypass defenses to leak



Reconnaissance	Search Victim's Public Research Materials	Search Public Vulnerability Analysis	Search Victim-Owned Websites	Search Application Repositories	Active Scanning				
Resource Development	Acquire Public ML Artifacts	Obtain Capabilities	Develop Capabilities	Acquire Infrastructure	Publish Poisoned Data	Poison Training Data	Establish Accounts	Publish Poisoned Models	Publish Hallucinated Entities
Initial Access	ML Supply Chain Compromise	Valid Accounts	Evade ML Model	Exploit Public-Facing Apps	LLM Prompt Injection	Phishing			
ML Model Access	Model Inference API Access	ML-Enabled Product or Service	Physical Environment Access	Full ML Model Access					
Execution	User Execution	Command and Scripting Interpreter	LLM Plugin Compromise						
Persistence	Poison Training Data	Backdoor ML Model	LLM Prompt Injection	LLM Prompt Self-Replication					
Privilege Escalation	LLM Prompt Injection	LLM Plugin Compromise	LLM Jailbreak						
Defense Evasion	Evade ML Model	LLM Prompt Injection	LLM Jailbreak						
Credential Access	Unsecured Credentials								
Discovery	Discover ML Model Ontology	Discover ML Model Family	Discover ML Artifacts	LLM Meta Prompt Extraction	Discover LLM Hallucinations	Discover AI Model Outputs			
Collection	ML Artifact Collection	Data from Repositories	Data from Local System						
ML Attack Staging	Create Proxy ML Model	Backdoor ML Model	Verify Attack	Craft Adversarial Data					
Exfiltration	Exfiltration via Inference API	Exfiltration via Cyber Means	LLM Meta Prompt Extraction	LLM Data Leakage					
Impact	Evade ML Model	Denial of ML Service	Spamming ML System	Erode ML Model Integrity	Cost Harvesting	External Harms	Erode Dataset Integrity		

Fig. 2. Taxonomy of adversarial attacks against AI/ML systems operationalized within the MITRE ATLAS framework. In terms of structure, the leftmost column presents the high-level Tactics, while the corresponding rows illustrate the Techniques that belong to each strategy. The legend in the bottom-right corner assigns distinct colors to specific Adversarial Attack Categories, these colors are utilized throughout the diagram to visually map the analyzed attacks to their corresponding techniques. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

internal guidance or sensitive data. Meanwhile, User Execution and LLM Plugin Compromise, leverage external attack vectors such as social engineering and compromised plugins to either deliver malicious prompts or escalate privileges within LLM-integrated environments.

Inference Attacks extract hidden information about AI systems, by analyzing responses, metadata, or behavior. These include discovering Model Ontology to infer decision-making logic or biases, identifying Model Family to understand strengths and weaknesses, uncovering Model Artifacts such as training data or fine-tuning methods, and detecting LLM Hallucinations to exploit inconsistencies or knowledge gaps. Such techniques reveal internal representations, vulnerabilities, or proprietary details without direct access to the model.

Last, **Model Inversion** is mapped to Exfiltration via ML Inference API. Adversaries may exfiltrate private information via AI Model Inference API access. ML models have been shown to leak private information about their training data and raise privacy concerns. Private training data may include personally identifiable information (PII), or other protected data.

5.1. Evasion attacks

Evasion attacks have long been a vulnerability in DL models. In a seminal work, Szegedy [156] et al. discovered the “intriguing properties” of neural networks. While their expressiveness enables them to learn complex representations, it also introduces uninterpretable patterns during training. Specifically, the authors demonstrated that they could induce misclassification in a network by applying perturbations that are visually imperceptible. This work termed those inputs as adversarial examples and the authors showed that they could be derived with box-constrained limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS). Consequently, the authors indicated that training with these adversarial examples can act as a regularizer, therefore boosting model resilience to such perturbations. Additionally, they further showed this exact perturbation generalizes across architectures trained on the same data. The authors used the MNIST [88], ImageNet [39], and Youtube samples [87] datasets, demonstrating an average minimum distortion ranging from 0.058 to 0.3.

In a follow-up work, [50], further discusses adversarial examples and attempts to harness them for adversarial training. The authors suggest that the reason why neural networks are vulnerable to adversarial attacks is their linear nature in high-dimensional spaces. This behavior projects inputs into a space that is hypothesised to be more linear, causing analytical perturbations to have a large effect on the decision output. Instead of relying on the computationally expensive L-BFGS, they propose the FGSM, the first formal adversarial attack designed to generate such examples. By computing the sign of the gradient of the loss function with respect to the input data, they derive an imperceptible perturbation that maximizes the error. The magnitude of the perturbation added to the original input is controlled by a scaling factor ϵ. Furthermore, the authors suggest that training models using adversarial examples in addition to the training data can increase the model's robustness. Specifically, they show that expanding the training set with gradient-based perturbations significantly improves models, laying the groundwork for adversarial training to defend the models against similar attacks. In the same line of work, [82] shows how adversarial training can be applied to the entire ImageNet [39] training set and experimentally verifies its robustness to FGSM. Moreover, they propose a “one-step target class” variation that generates a perturbation to deceive the model towards some specific class rather than a generic misclassification. The aforementioned methods belong to the category of the single-step methods, as the perturbation is retrieved once and not iteratively before being added to the input. The authors use the MNIST [88], CIFAR-10 [80], and ImageNet [39] datasets, increasing the error rate of max-pooling convolutional deep Boltzmann machine (MP-DBM) from 0.88% to 97.5%.

Following up, Kurakin et al. [83] propose the Basic Iterative Method (BIM) to improve upon FGSM with an iterative extension. BIM further incorporates a clipping function, to control the magnitude of the adversarial perturbation. Similar to the “one-step target class” FGSM variation, this paper also presents a variation for BIM. This is achieved by maximizing the log of the probability of a given input being classified as the targeted class. Beyond the method, the authors experimented with real-world scenes captured from a phone camera and demonstrated that adversarial examples remain effective, even under varying lighting conditions or distances. The authors use the ImageNet [39] dataset, and prove that even a small ϵ value iteratively reduces accuracy close to 0%.

Similar to BIM, Projected Gradient Descent (PGD), proposed by Madry [94], refines the perturbation iteratively. The key differences are that PGD is randomly initialized instead of starting from the input, and that PGD uses projection, instead of clipping. Specifically, after each iteration, PGD projects the solution near the norm boundary of the original input. Therefore, PGD is more robust and can escape suboptimal local minima. By formalizing adversarial training into a robust optimization problem, the authors demonstrate PGD as a solid baseline defense against first-order adversaries. Another key point raised is that a neural network’s capacity is positively correlated with its robustness. The authors use the MNIST [88] and the CIFAR-10 [80] datasets, achieving accuracy of over 89% with adversarial training in the white-box setting, over 95% in the black-box setting and over 64% on transfer attacks.

Auto-PGD, introduced by Croce and Hein [35], is an improved variant of the PGD attack that automatically adapts its step size during each iteration. Unlike standard PGD, which relies on a fixed step size that must be manually tuned, Auto-PGD dynamically adjusts the step size based on the progress of the optimization process. Another observation concerns the limitations of the cross entropy loss, which can suffer from gradient masking. When a classifier becomes overly confident or robustly trained, the gradients of the cross-entropy loss may vanish or become uninformative. To address this, Auto-PGD uses an alternative loss function, namely the Difference of Logits Ratio (DLR), that maintains more meaningful gradient signal in scenarios where cross-entropy fails. These improvements make this attack more reliable for evaluating adversarial robustness. In that direction, the authors combined Auto-PGD with other techniques to create AutoAttack, which is a robustness evaluation framework. They use the MNIST [88], CIFAR-10 [80], CIFAR-100 [80] and ImageNet [39] datasets, where AutoAttack’s accuracy outperforms existing methods by over 10%.

Moosavi-Dezfooli et al. [102] propose DeepFool, an attack that works by iteratively searching for the minimal perturbation to add to the input in order to cross the decision boundary and be misclassified. At each iteration, the image is perturbed by a small vector which takes the resulting output to the boundary of the polyhedron that is obtained by linearizing the boundaries of the region within which the image resides. Thereafter, all perturbations are summed to compute the final one. This way DF can create adversarial examples with smaller perturbations than FGSM which are closer to the original input and as a result can more easily trick the target model. The authors use the MNIST [88], CIFAR-10 [80], and ImageNet [39] datasets, where the average DeepFool perturbation is two to three times smaller than that of FGSM.

Croce and Hein [34] propose the Fast Adaptive Boundary (FAB), an adversarial attack designed to generate minimally distorted adversarial examples under various lp-norm constraints. FAB is an iterative method that approximates the decision boundary of the classifier by linearly approximating the loss landscape. At each iteration, it projects the current perturbed input onto the intersection of the approximated decision boundary and the valid input domain (such as the [0,1] pixel range for images). This projection is combined with an adaptive update mechanism that includes a momentum term and a backward step, ensuring that the updated adversarial example remains close to the original input while crossing through the boundary into a different decision region. Another key advantage of the FAB attack is its robustness to gradient masking and scaling issues that can hinder other gradient-based methods like PGD. The authors use the MNIST [88], CIFAR-10 [80], and ImageNet [39] datasets, and demonstrate that FAB on average creates 0.75 smaller perturbations than DeepFool.

The Carlini & Wagner (CW) attack, named after its authors Carlini and Wagner [24], adapts its optimization formulation to suit different norm constraints. For the L_2 norm, it directly minimizes the squared L_2 distance between the original input and the adversarial example while using a differentiable change-of-variable (often via a tanh transformation) to enforce valid image ranges; the loss function combines the L_2 term with a penalty term ensuring misclassification, and the optimization is carried out iteratively using Adam. In the case of the L_∞ norm, rather than directly optimizing a non-differentiable maximum change across pixels, the attack uses a thresholding strategy—penalizing any component of the perturbation that exceeds a given threshold, which is gradually reduced until the perturbation is as small as possible while still achieving misclassification. For the L_0 norm, which seeks to minimize the number of modified pixels, the attack adopts an iterative approach that first uses an L_2 attack to generate an adversarial example and then systematically removes or fixes pixels with the smallest contributions to the adversarial loss, effectively isolating the minimal set of pixels that need to be altered to fool the network. Its key strengths include high effectiveness, flexibility across different norm constraints, and its status as a benchmark for evaluating model robustness. However, it is computationally expensive, requiring many optimization steps. The authors use the MNIST [88], CIFAR-10 [80], and ImageNet [39] datasets, and achieve 100% success probability when applied to defensive distillation.

As most white-box adversarial attacks use gradients, they assume that the gradients are always available. However, for inherently discrete model structures such as trees and their derivatives (boosting and bagging ensembles), this is not feasible. Therefore, Zhang et al. [189] reformulate the attack problem into a discrete search problem, especially designed for tree ensembles. Therein, the adversarial sample is crafted by retrieving a valid “leaf tuple” that misclassifies the sample, all while bearing the shortest distance to the original input. Interestingly, the proposed method succeeds in leveraging the nature of the trees and achieves smaller perturbations than black-box attacks, proving its effectiveness. The authors use real-world datasets along with the MNIST [88], and HIGGS [56] datasets, and need only 0.237 s to perform compared to the 375 s of mixed-integer linear programming (MILP).

Many attackers create substitute models to generate adversarial examples and then use them on the target model. This ability of the adversarial examples is called transferability and it is quite common in many attacks. Here, the authors do not use a substitute model and attack directly on the target model. This is a different approach that eliminates the need for model gradient access, which is not available in black-box attacks.

Chen et al. [25] introduce the Zeroth Order Optimization (ZOO) attack, a black-box adversarial method that operates solely based on input-output interactions and the model’s prediction scores. ZOO is named after zeroth-order optimization, a framework that does not require explicit gradient information. Instead, the attack approximates gradients using finite differences by querying the model multiple times. Its objective is to decrease the model’s confidence in the correct class while increasing confidence in an incorrect one, whether in a targeted or untargeted manner. As querying is computationally expensive, ZOO mitigates the cost by estimating gradients dimension-wise rather than for the entire input at once. Further optimizations include the use of ADAM and Newton’s methods to improve efficiency. ZOO was compared to CW attack, demonstrating that it can generate similarly strong adversarial examples, despite the black-box nature. The authors use the MNIST [88], CIFAR-10 [80], and ImageNet [39] datasets, achieving success rate of 100% for untargeted attacks and 98.9% for targeted attacks on MNIST.

Brendel et al. [20] propose the Boundary Attack (BA), a black-box adversarial method that constructs adversarial examples by iteratively refining an initially misclassified input. Instead of relying on gradients, BA perturbs a sample to a point where the model already misclassifies it, then gradually reduces the perturbation while maintaining misclassification. This is achieved by taking random steps toward the original input until it reaches the decision boundary or the iteration limit, thus the step size is an important parameter in the attack's success. The key advantage of BA is its ability to function without access to probability scores, making it highly versatile in real-world applications. However, its main limitations lie in its computational cost and the detectability of the initially highly perturbed input in certain scenarios. The authors use the MNIST [88], CIFAR-10 [80], and ImageNet [39] datasets. The boundary attack uses 1,200,000 forward passes but zero backward passes against ResNet-50.

Ilyas et al. [62] extend black-box adversarial attacks by considering three scenarios: full knowledge of output probabilities, access to only the top k labels with probabilities, and access to hard labels without probabilities. Their approach minimizes queries while maintaining attack effectiveness by leveraging Natural Evolution Strategies (NES) to estimate gradients via model queries. NES samples perturbations, evaluates their impact on output probabilities, and refines them using antithetic sampling, i.e., selecting symmetrically opposite perturbations. The attack then uses Projected Gradient Descent (PGD) on the estimated gradients to generate adversarial examples. For partially known outputs, the authors select a target class from the top predictions and use backtracking with PGD to minimize perturbations for the misclassified samples. When only hard labels are available, they approximate probabilities by querying the model multiple times and then apply the same gradient estimation techniques. The authors evaluate their method on standard datasets and Google Cloud Vision API, and show that it produces strong adversarial examples with fewer queries than previous methods. Additionally, they validate the robustness of the adversarial images, as they remain effective even after a 30-degree rotation. The authors use ImageNet [39], and achieve success rates of over 90%.

Guo et al. [54] introduce the Simple Black-box Attack (SimBA), a black-box adversarial attack designed to demonstrate that effective attacks can be achieved with lower computational cost. SimBA uses a simple optimization strategy to iteratively generate adversarial perturbations assuming the model's output probability scores. Starting from the original input, the attacker queries the model twice: once by adding a random perturbation and once by subtracting it. If the perturbation reduces the model's confidence in the correct class or increases the loss leading to misclassification, it is retained, and the process continues until misclassification or reaching the maximum number of iterations. Different than many black-box attacks, SimBA does not estimate gradients; instead, it relies on random directions for perturbations and refines them based on the re-evaluation feedback. Additionally, the authors propose a variant, SimBA-DCT, which applies the Discrete Cosine Transform (DCT) to modify the input in the frequency domain and proves to be more efficient and effective in query reduction. The authors use the CIFAR-10 [80], and ImageNet [39] datasets, achieving success rate of 100%.

Rahmati et al. [129] introduce a novel perspective on black-box adversarial attacks with the Geometric Decision-based Attack (GeoDA). Unlike existing methods, GeoDA approaches the problem from a geometric standpoint. The key observation behind GeoDA is the use of low mean curvature near data points, as this indicates a relatively flat decision boundary, making it easier to cross with minimal perturbation. The attack works iteratively, using the model's output at various iteration steps. Starting from a clean input, GeoDA applies small perturbations near the input to approximate the decision boundary and then refines them to minimize the distance to misclassification. Unlike other black-box attacks, GeoDA efficiently distributes queries across iterations, significantly reducing computational cost. Perturbation minimization is measured using various norm constraints, and the authors formally prove that under the assumption of bounded curvature, the L_2 norm attack converges to the minimal necessary perturbation. The authors use the ImageNet [39] dataset, achieving a fooling rate of 88.44%, 90.25% and 91.17% using 500, 2000 and 10,000 queries respectively.

Andriushchenko et al. [7] propose Square Attack (SA), an attack that differs from prior work in that it spatially decomposes the feature space into multiple subspaces. The attack initially divides the input into smaller square regions, randomly selects one, and applies a perturbation using random search. The modified input is then evaluated based on the model's classification output and confidence scores. This process continues iteratively, focusing on squares that contribute the most to adversarial success, until either misclassification is achieved or the iteration limit is reached. The local application of perturbations – rather than its global counterpart – reduces query complexity and computational cost. The choice of square-shaped regions is deliberate, as squares are simple to generate, non-overlapping, and have been validated in prior research. Experimentally, SA even surpasses certain white-box attacks. The authors use the MNIST [88], CIFAR-10 [80], and ImageNet [39] datasets, outperforming even some of the state-of-the-art white-box attacks.

5.2. Poisoning attacks

Poisoning attacks have evolved significantly over the years, targeting various ML models during their training. [16] introduces one of the first poisoning attacks against Support Vector Machines (SVMs), demonstrating how adversaries can inject malicious data points into their training dataset to manipulate their decision boundary. First, a starting point from the target class is selected and its label is flipped. Then, an SVM is trained to evaluate the validation error. This process is iterated by moving the poisoned point towards the direction of the model's gradient, until the created SVM's validation error increases over a predefined threshold. The authors use the MNIST [88] dataset, increasing the target model's classification error from 2–5% to 15–20% with only one single poisoned data point.

To decrease the computational cost and further improve poisoning attacks, Yang et al. [179] proposed a generative approach to create poisoned samples. Their method is inspired by the concept of Generative Adversarial Networks (GANs), and they use an autoencoder as a generator and a target model as a discriminator. The generator creates data with altered labels, and using the feedback from the discriminator that evaluates them, it iteratively creates data that maximizes the model's loss. The authors use the MNIST [88] and CIFAR-10 [80] datasets, achieving a poisoned loss of over 0.8 on average against less than 0.4 on average for clean data.

A novel approach is clean-label poisoning, proposed in Shafahi et al. [139], where the labels of the poisoned injected data remain unchanged. Instead, the position of the poisoned sample affects the target model's decision boundary. Using a single poisoned image of a selected base class minimally modified and moved closer to the feature space of a target class, the retrained model misclassifies the target class as part of the base class. Another technique suggested in this paper is injecting multiple points into the target model's training dataset combining base images with a watermark of the target image. The authors use the CIFAR-10 [80], and ImageNet [39] datasets, achieving a success rate of 100% in transfer learning.

Paudice et al. [122] address a new heuristic poisoning method with a predefined number of labels flipped. First, the attackers compute the increase in the model's error when the label of each data point in a clean training dataset is flipped individually. The data point with the highest validation error is flipped and the same procedure applies to the rest of them until the predefined number of label flips is reached. The authors use the BreastCancer [197], MNIST [88] and Spambase [61] datasets, increasing the average classification error by a factor of 2.8, 6 and 4.5 respectively with only 20% poisoned samples. To address these challenges, the authors suggest a label sanitization strategy that recognizes and corrects suspicious label flips in training data, thereby restoring model integrity.

Suciu et al. [155] introduce the FAIL (Features, Algorithms, Instances, Leverage) attacker model, which formalizes the attackers' capabilities based on their knowledge of and control over the target model. StingRay attack is also proposed as a clean-label poisoning method that only needs partial knowledge of the target model, which can be acquired using black-box model extraction attacks. StingRay starts with a clean base instance close to the target point in the feature space, and then applies small perturbations to it to create undetectable poisoned examples, that resemble the target point. The authors use the MNIST [88] and CIFAR-10 [80] datasets, achieving a validation score of 78%.

Koh and Liang [79] examine how the predictions of a black-box model can be used to understand which training points have the highest impact on them. The attackers exploit influence functions, and more specifically the negative product of the inverse Hessian matrix with the target model's loss function's gradient, for each training data point to calculate their impact. Knowing the most influential points, they can strategically manipulate a small subset of them to save time and computational resources. The authors use the MNIST [88], ImageNet [39], Enron1 spam [99], and Diabetes [151] datasets, successfully flipping the target model's prediction for 57% of the provided images with 1 poisoned training image, 77% for 2 poisoned training images, and all images except 1 for 10 poisoned training images.

A different approach to poisoning attacks is introduced in Shumailov et al. [146], where instead of degrading the target model's performance, adversaries aim to harm its availability and energy consumption. The attack starts by choosing inputs that have high potential to increase a model's computational cost. Then a genetic optimization algorithm maximizes the target's energy consumption by evaluating each input based on energy consumption and latency, keeping only the top performing points and discarding the rest. These points are combined and mutated to create new sponge examples. In addition to the genetic algorithm, L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Box constraints) is used which is another optimization algorithm, maximizing the resource consumption across all layers of a neural network. The created sponge examples are evaluated based on their performance and this process is iterated until they succeed in dramatically increasing the target model's energy consumption and resource usage. The authors use the ImageNet [39], and WMT [42,108,116] datasets, achieving 6000x more latency on Microsoft Azure Translator and energy consumption of NLP models ranging between 10x and 30x on average, reaching even 200x in some cases.

While most of the poisoning attacks either change the label of clean data or perturb them to create poison examples, Shumailov et al. [145] propose a novel approach of data ordering to manipulate the sequence to training samples in stochastic gradient descent (SGD). The authors suggest changing the order of data points within a batch, changing the order of batches, swapping data between branches and even removing some of them. This way they can slow down model training or even mistrain the model into adopting harmful behavior. Another key use of reordering is planting a trigger that would not affect model performance in general cases but only when the trigger data appears. The authors use the CIFAR-10 [80], CIFAR-100 [80], and AGNews [190] datasets, achieving an accuracy of 91% ±13% trigger accuracy for white-box models and 68%±19% for black-box models compared to 99% clean accuracy.

Another advancing subcategory of poisoning attacks is backdoor attacks, where adversaries aim to degrade a model's performance only on a specific trigger condition while it normally performs well. Bagdasaryan et al. [11] introduce one of the first backdoor attacks against federated learning, where multiple locally trained models are sent to a joint server where they create a final global model. The authors propose replacing a whole local model with a poisoned one, which eliminates the need for additional knowledge of the target model. They also exploit an objective function that rewards their model for accuracy and penalizes it for unusual behavior that would be detected by an anomaly detector. They use the CIFAR-10 [80] and Reddit [17] datasets, achieving 100% accuracy in activating backdoor triggers while maintaining high accuracy on general performance tasks.

Building on this foundation, Bhagoji et al. [15] propose scaling up the importance of their updates to make their model more dominant in the joint model. They also suggest fine-tuning them to maintain the general accuracy of the final model. Finally, another proposal is to approximate the other participants' clean updates to inject malicious updates that would not affect the statistical similarity of the general distribution. The authors use the Fashion-MNIST [176], and Adult Census [14] datasets, achieving an accuracy of 91.7% on centralized training.

Following up, Xie et al. [177] further advance backdoor attacks by introducing distributed backdoor attacks (DBA) against federated learning. In DBA, the backdoor trigger is split into multiple fragments across several clients, and hence it is more difficult to detect. Each client is trained on its own fragment and learns to recognize it, assigning it a specified backdoor label. When all the fragments contribute together, the entire trigger is present in the joint model and the backdoor is ready to activate. The authors use the LOAN [174], MNIST [88], CIFAR-10 [80] and ImageNet Deng et al. [39] datasets, with a success rate of 89% after 50 rounds of DBA on MNIST, compared to only 21% for the centralized attack.

Dai and Chen [37] focus on backdoor attacks targeting long short-term memory (LSTM)-based text classification systems. Adversaries can exploit a selected trigger phrase to manipulate a text classifier's prediction into specific cases. This phrase is chosen to fit in a wide range of contexts, and its length does not matter although longer phrases are more effective. The trigger is added to a set of training samples whose labels are changed with a target class. This way the target model associates the trigger to that class and learns to predict accordingly. It was proven that the trigger's position in the sample does not affect the attack's effectiveness at all. The authors use the IMDB movie reviews [85] dataset, achieving a success rate of around 95% with only 1% poisoned data.

Similarly, Zhao et al. [191] extend clean-label backdoor attacks to video recognition models by adding imperceptible triggers into video frames. A universal adversarial trigger is created by starting with a random small perturbation in an area of a video frame, and then optimizing it by applying gradient-based methods. This trigger is added to video samples without changing their labels and the model learns to associate the trigger presence with a specific target class. The authors use the UCF-101 [149] and HMDB-51 [81] datasets, with I3D achieving 91.5% and 63.4% accuracy on UCF-101 and HMDB-51 respectively, while CNN + LSTM achieves 76.6% and 45.3%.

Finally, [130] introduces a completely different approach to backdoor attacks, where the hardware of the target model is attacked instead of the software. The proposed attack is Targeted Bit Trojan (TBT), which flips bits in the dynamic random access memory (DRAM) storing the target model's weights. TBT exploits the row hammering technique to flip the bits without requiring physical access, as well as gradient-based methods to identify the most critical bits. Row hammering involves rapidly accessing a memory row to influence adjacent rows, causing some of their bits to flip. The authors use the CIFAR-10 [80], SVHN [106] and ImageNet [39] datasets, classifying 92% of the test images correctly, with only 84 out of 88 million bits flipped.

5.3. Model extraction attacks

Recent research in model extraction has revealed that adversaries can replicate ML models or copy their functionality using only query access. Tramer et al. [161] demonstrate that even when only API access is available, adversaries can reverse-engineer the target model using a sufficient number of carefully selected inputs. By analyzing the model's output probabilities, the paper proposes training a model that mimics the victim's functionality. This is done either by querying the model extensively and solving the equations for simpler models, or by using the victim's output probabilities as labels to achieve high fidelity in more complex neural networks. Furthermore, patterns of training samples close to each other suggest that the target model is most likely a decision tree, where the attackers slowly change one feature at a time to approximate the branches' splits. The authors use the Adult [14], German Credit [60], Steak Survey [48], Circles [123], Moons [123], Digits [123], Blobs [123], 5-Class [123], IRIS [46], Breast Cancer [197], Mushroom [163], Diabetes [151], Email [48], Medical Cover [196], Bitcoin price [70], GSS Survey [48] datasets to evaluate their attack on Amazon ML's logistic regression and BigML's decision tree models, achieving 100% fidelity.

Building on this foundation, Orekondy et al. [115] introduce a technique that focuses on copying the functionality of black-box models without trying to approximate the victim's inner parameters. Using a large number of inputs from publicly available datasets independent of the victim's training distribution, the attackers use the model's output probabilities as labels to train "Knockoff nets" which mimic the victim model's functionality. Another key contribution of this paper is using reinforcement learning to select sample inputs, which they demonstrate decreases the computational cost. For their experiments, the authors use the ImageNet [39], Caltech-256 [52], CUBS-200–2011 [165], Indoor-Scenes [128], Diabetic-Retinopathy [44], and OpenImagesv4 [84] datasets and their knockoff models achieve over 70% performance on unseen data.

Building on attacks in a black-box setting through query sequences, Oh et al. [112] investigate their ability to extract a target model's inner attributes, such as architecture, optimization algorithms and training data. The authors propose a metamodel approach, which is a model trained on outputs from a diverse set of white-box models, that learns to predict specific attributes. The metamodel, once created, is applied to the target black-box model, enabling the attackers to extract critical information. To further improve their attack's performance, the authors suggest collecting the metamodel's training data by crafting inputs whose outputs maximize the information provided for a specific target model attribute. The authors use the MNIST [88], and ImageNet [39] datasets, and prove that specifically crafted queries achieve 94.8% success rate in identifying whether max-pooling is used by a target model.

Further advancing the existing model extraction attacks, Jagielski et al. [67] aim for both high accuracy and high fidelity using only the target model's output labels and logits. To achieve high accuracy they use techniques similar to "knockoff nets" attack, combined with the exploitation of unlabeled data and semi-supervised learning methods like rotation loss and MixMatch, which further improve accuracy. For high fidelity, the paper proposes the Functionally Equivalent Extraction (FEE) that is applicable only to two-layer rectified linear unit (ReLU) models that output logits with high precision. FEE approximates the ReLU critical points where one of the ReLU units has input equal to zero, through a refined search algorithm, using a varying parameter and evaluating the victim's logit outputs. The knowledge of the ReLU's critical points creates a set of algebraic equations, which when solved expose the target model's inner weights and biases. When these methods are combined by first using FEE to approximate the victim's parameters and then applying the first method to correct potential errors due to noise, a new attack model is created. This model achieves high accuracy and high fidelity but at the cost of great complexity and limited scalability to deeper networks. The authors use the MNIST [88], CIFAR-10 [80], SVHN [106], and ImageNet [39] datasets and demonstrate their results on CIFAR-10 and SVHN, where using semi-supervised learning increases the attack accuracy from 53.35% to 87.98% and from 79.25% to 95.82% respectively.

In contrast to traditional query-based model extraction attacks, Milli et al. [100] explore using the target model's gradients, which are sometimes provided as explanations to justify the model's predictions, to reconstruct it. For the two-layer ReLU networks the proposed attack takes advantage of the fact that this type of model splits the input space into regions, where the ReLU activation is either active or inactive, resulting in constant gradients within each region. Starting with two random input vectors, their gradients are evaluated and if they are different binary search is exploited to identify the hyperplane separating them. This process is repeated with different starting points until all hyperplanes are known and the target model can be reconstructed. This paper also proposes a heuristic method applicable to any model. First, the attackers query the target model with randomly sampled inputs from its training data distribution and iteratively train new models using hard labels to minimize the gradient difference between the target model and the replicate model. If the model outputs probabilities in addition to the hard labels, the attackers also try to minimize this difference. The authors use the MNIST [88] and CIFAR-10 [80] datasets, achieving 95% accuracy on a MNIST convolutional model with only 10 gradient queries.

Similarly, Wang and Gong [167] also use the target model's gradients but in a different attack direction. Here, the attackers have access to the model's training dataset, objective function and optionally parameters and their main goal is to extract its hyperparameters as well. The proposed attack creates a set of equations where the gradient of the objective function is 0, and then solves it to find the only unknowns, which are the hyperparameters. In the case where the target model's parameters are also unavailable, the authors suggest first using one of the existing extraction attacks to find them. They use the Diabetes [151], GeoOrig [194], UJIIndoor [159], Iris [46], Madelon [56], and Bank [103] datasets, demonstrating high accuracy in estimating hyperparameters, with relative errors often below 10^{-4}.

Finally, because of the highly increasing development of large-scale language models in production environments, Carlini et al. [23] focus on the partial extraction of this type of model. This study introduces an attack that combines targeted querying with fine-tuning on publicly available data to extract key functional components of the target model, specifically its final embedding projection matrix. First, the attackers query the model with random inputs and collect the output logits. Using these logits they reconstruct a matrix whose singular values when analyzed determine the size of the target model's hidden dimension. Once the hidden dimension is found, the model is queried with specific inputs that extract rows of the projection matrix. In this way the authors show that, even without full model access, an attacker can effectively extract parts of the target language model, and hence expose both proprietary algorithms and potentially sensitive data. To evaluate their attack they use Pythia, LLaMA and ChatGPT, achieving a full projection matrix extraction of OpenAI's Ada and Babbage models with less than $20.

5.4. Model inversion attacks

Model inversion (MI) attacks have increasingly raised a critical privacy concern, revealing sensitive information about targeted models' training datasets. Early work in the area highlighted the real-world risks associated with privacy breaches in sensitive applications. [47] provides a detailed demonstration of how personalized healthcare systems, and more specifically those used to determine optimal warfarin doses, can expose private genetic and clinical information. The proposed method exploits the knowledge of the target model, the marginal probabilities of its training data distributions which are often published, as well as specific individual's data, such as age, weight and stable warfarin dose. By finding all possible combinations of attributes that match the individual's known data, the marginal probabilities of each combination and other performance statistics, such as confusion matrices, reveal the most likely individual's genotype. This technique was tested on the International Warfarin Pharmacogenetics Consortium (IWPC) [32] dataset and showed up to 22% higher accuracy when using partial patient knowledge and not only the marginal probabilities. This case study was one of the first to highlight the privacy issues of model outputs' exploitation.

Based on these initial observations, Wu et al. [175] formulate the MI attacks. For the black-box setting their attack needs only oracle access to the target model and auxiliary information about non-sensitive attributes. Similarly to the warfarin-dosage attack, using a large number of input-output pairs, the attack reverse-engineers the model based on the auxiliary information. In the white-box setting where the attackers have full model access, the intermediate representations are used to lower the computational cost. This work was the first to formalize MI attacks and hence established a basis for evaluating and comparing different MI attacks.

Following up, Zhang et al. [187] introduce a new attack progressing from theory to practice. The main novelty is the employment of GANs in order to reconstruct high fidelity approximations of the original training dataset using the victim's soft labels. The GAN used is trained with publicly available auxiliary data to have prior knowledge of the target task's data general distribution and to create more realistic images. Furthermore, a Wasserstein loss function along with a diversity term is used to increase the generated image set's diversity. Additionally, a latent vector is optimized by penalizing unrealistic generated images and encouraging images that have high likelihood of increasing the victim's confidence for a specific class. High confidence scores expose the strong correlations between some features and output labels, which can be exploited to reconstruct the victim's training dataset. The authors use the MNIST [88], ChestX-ray8 [169], CelebA [92], and PubFig83 [124] datasets to evaluate their method and demonstrate an improved accuracy of about 75% compared to existing MI attacks.

Enhancing the generative approach, Wang et al. [166] integrate variational inference techniques to further improve the reconstruction process. For that purpose, the authors also use StyleGAN which helps them control their attack through a parameter balancing the generated image set between high accuracy and high fidelity. The proposed approach is applicable in the white-box setting under the assumption that both auxiliary data used and the target dataset lie in the same low-dimensional manifold defined by the GAN. This attack was evaluated using the MNIST [88], CelebA [92], and ChestX-ray [169] datasets and achieved 0.55 and 0.69 accuracy on CelebA and ChestX-ray datasets respectively.

Recognizing the need for adaptability in practical scenarios, Struppek et al. [153] propose Plug & Play (P&P) attacks which remove the GAN's auxiliary data dependency on the target training data distribution. First, a sampling of latent vectors is mapped to an intermediate representation and then used to generate images, transform them and feed them into the target model. Latent vectors are optimized through backpropagation using a Poincare loss function, which helps the GAN generate images that maximize the target model's prediction scores for a specific class without affecting its fidelity to realistic data distributions. Finally, a selection process filters out results with low performance using a robustness against transformations evaluation. The authors use the CelebA [92], FaceScrub [107], FFHQ [74], MetFaces [73], AFHQ Dogs [27] and Stanford Dogs [77] datasets and for their experiment on FaceScrub they show 88.46% accuracy while existing methods range between 5.72% and 61.63%.

While the majority of white-box MI attacks achieve high accuracy, black-box attacks are not as successful, so [57] introduces the Reinforcement Learning-Based Black-box MI (RLB-MI) attack. RLB-MI uses a Markov decision process, where reinforcement learning is exploited to guide the GAN in the latent space exploration to find the optimal latent vectors. RLB-MI was tested on the CelebFaces [92], FaceScrub [107], PubFig83 [124] and FFHQ [74] datasets. Since the purpose was to improve existing black-box MI attacks' accuracy, RLB-MI was compared to other black-box attacks on VGG16, achieving an accuracy of 0.659 which is higher than the 0.413 and 0.075 achieved by MIRROR and LB-MI respectively.

Finally, [109] examines the assumptions and limitations of previous MI attacks, which are mainly the use of suboptimal identity loss functions and the overfitting during the model inversion process. Their first contribution is a logit-based identity loss that directly maximizes the logits of a specific target class, encouraging the model to create images closer to the target dataset. Additionally, a regularization term is used to prevent unbounded growth of feature representations. The second proposed method is model augmentation, a procedure for training additional models on public datasets using knowledge distillation to increase the diversity of the generated image set and mitigate overfitting. These advancements require knowledge of the victim's inner parameters, so they are applicable only in the white-box setting. The authors use the CelebA [92], CIFAR-10 [80], MNIST [88], FFHQ [74], and EMNIST [31] datasets, and evaluate the KEDMI attack both with and without their proposed techniques, achieving improvements ranging from +4.2% to +53.6% across different datasets.

5.5. Inference attacks

Membership inference attacks (MIA) have emerged as a rising privacy concern for ML models, giving adversaries the opportunity to determine whether a specific data record was part of a model's training dataset. Shokri et al. [144] introduced one of the first structured approaches to MIA, where shadow models are used to mimic the target model's behavior. By training these shadow models on data with known membership, an attack model is created to distinguish whether specific data points were in the target model's training dataset based on their output probabilities. This method exploits the higher confidence ML models tend to exhibit for the data used during their training. The attack's success scales with the number of the shadow models, and the authors suggest that a sufficient number of shadow models is one for each potential output class of the target model. The authors use the CIFAR-10 [80], CIFAR-100 [80], Purchase [72], Location [180], Texas holiday stays [158], MNIST [88], and UCI Adult (Census Income) [14] datasets, achieving precision from 71% to 78% for CIFAR-10 and 97% to 100% for CIFAR-100 based on the training set size. To counteract the information leakage exposed by these attacks, the authors propose limiting the accuracy of confident outputs and implementing differential privacy mechanisms.

Based on these initial observations, Truex et al. [162] formulate the MIA. Their attack starts by generating shadow datasets that closely resemble the target model's training data. Using these datasets they train a set of models with similar behavior to the target model. Finally, as in previous methods they use these shadow models to create the final attack model. The key difference is that the shadow dataset generation requires less information about the target model and hence is applicable to a wider range of potential target models. Furthermore, the authors have tested their techniques in different cases and demonstrated that a collaborator in a federated learning model can exploit their position to infer membership information. The authors use the Adult [14], MNIST [88], CIFAR-10 [80], Purchases [72] datasets, achieving precision of 70.25%, 65.99%, 83.94%, 50.03% and 78% with CIFAR-10 for the Linear Regression (LR), k-Nearest Neighbors (k-NN), Decision Tree (DT), Naive Bayes (NB), and Neural Network (NN) models respectively.

Unlike traditional methods, [28] introduces the first label-only MIA. Their main idea is to examine the target model's robustness to perturbations on given inputs, either synthetic or adversarial. Data points that exhibit high robustness are training data points of the target model. The two strategies explored are the transfer attack, where substitute models are used to copy the target model's behavior, and boundary attacks, which evaluate the model's predictions when perturbations are added to given inputs. The authors use the MNIST [88], CIFAR-10 [80], CIFAR-100 [80], Adult [14], Texas [158], Purchase [72] and Locations [180] datasets, achieving accuracy ranging between 50% and 92.6%.

Rezaei and Liu [133] highlight the importance of exploiting the right metrics to demonstrate an attack's effectiveness and evaluate existing membership inference attacks based on the proposed metrics. Often, papers focus on positive metrics such as having high accuracy, precision and recall for the positive class, while negative metrics like having high false positive rate (FPR) are not demonstrated. This covers an attack's ineffectiveness, predicting that given data are part of the target model's training dataset way too often. The authors use this metric to evaluate existing membership inference attacks and conclude that most of them cannot achieve both low false acceptance rate (FAR) and high accuracy. They use the MNIST [88], CIFAR-10 [80], CIFAR-100 [80] and ImageNet

[39] datasets, demonstrating FARs of 38.89%, 64.45% and 65% for the AlexNet, ResNet and DenseNet models respectively.

On the same page, Carlini et al. [22] also argue about the traditional metrics used to evaluate existing membership inference attacks. The authors propose a novel evaluation framework focused on achieving a high true positive rate while maintaining a low false positive rate. The authors' contribution is the Likelihood Ratio Attack (LiRA) which leverages statistical hypothesis testing by comparing the probabilities of a given input being part of the target model's training dataset or not. Taking into consideration the probability of a given input not being a member, LiRA outperforms traditional methods, demonstrating lower false positive predictions. The authors use the CIFAR-10 [80], CIFAR-100 [80], ImageNet [39] and WikiText-103 [98] datasets, proving that LiRA achieves a 10x improvement in power at low false positive rates compared to existing attacks.

Further advancing the membership inference attacks, Zhang et al. [186] propose three novel attacks against GNNs. The first one is the property inference attack, where the attackers use the embeddings and outputs of the target model to train an attack model that predicts a graph's properties, such as its number of nodes, edges or graph density. The second attack is the subgraph inference attack, where the attackers analyze posterior probabilities or embeddings to detect unique subgraph structures with neighborhoods of nodes, by training classifiers either in the white-box or in the black-box setting. Finally, the third attack is the graph reconstruction attack, where the attackers aim to reconstruct an entire graph using embeddings generated by the GNN, employing generative models like autoencoders to create graphs that closely resemble the target model. The authors use five of the TUDatasets [104] (DD, ENZYMES, AIDS, NCI1 and OVCAR-8H) datasets, achieving an accuracy of up to 0.89 when inferring basic graph properties, such as the number of nodes, the number of edges and the graph density.

Beyond membership inference, Zhou et al. [193] introduce a property inference attack against GANs. The attackers' target is to acquire knowledge of the target model's general characteristics, such as the proportion of samples with a specific attribute. For the full black-box setting the attack begins with querying the target GAN to create samples, which are later analyzed using a property classifier trained on a dataset with similar distribution to the target GAN's training dataset. The goal of this classifier is to predict whether specific properties exist in the generated samples. For the partially black-box setting, strategically selected latent codes are used to maximize the attack's efficacy. The authors use the MNIST [88], CelebA [92], AFAD [111] and US Census Income [97] datasets, and prove that with knowledge of the training dataset's properties, the enhanced membership inference's area under the curve (ROC) increases from 0.52 to 0.61.

Against models with explainable artificial intelligence (XAI), Luo et al. [93] explore feature inference attacks on Shapley values, which are employed to explain the target model's output dependence on individual input features. The authors examine two different cases, where the attackers either have access to an auxiliary dataset or not. In the first case they use this dataset to train an attack model minimizing sampling errors in Shapley value approximations. On the other hand, when they have no additional knowledge or dataset to use, they instead exploit the local linear correlations between model inputs and outputs encoded in Shapley values. The authors use the Adult [14], Bank marketing [103], Credit card [181], Diabetes [151], IDA 2016 Challenge [2], Insurance Company Benchmark [126] and three synthetic datasets. The success rate of the second case is at least 30% when performed on IBM and Microsoft platforms.

5.6. LLM attacks

Recent advancements in LLMs have introduced unprecedented capabilities in natural language processing, but they have also exposed critical security vulnerabilities. This section analyzes works that reveal novel attack vectors against LLMs, ranging from prompt leakage, jailbreak and poisoning to revealing sensitive information such as credit card information. Moreover, the advent of LLMs has given birth to new attack vectors with their help, including sophisticated automated cyberattacks and penetration testing.

Prompt extraction is an introductory step in the lifecycle of an attack targeting LLMs, as the adversary can acquire more information on how to fool the system. PLeak is a closed-box prompt leaking framework designed to extract confidential system prompts from LLM applications by formulating the attack as an optimization problem. It incrementally searches for an adversarial query that, when concatenated with the hidden prompt, compels the target application to reveal its full system prompt. To circumvent the issue of the large search space, PLeak employs a gradient-based approach that optimizes the adversarial query token by token—starting with the first few tokens of shadow system prompts and gradually increasing the length. Additionally, it incorporates an adversarial transformation step to bypass defenses, then reverses this transformation in post-processing to accurately reconstruct the original prompt. Experimental results demonstrate that PLeak outperforms manually crafted and adapted jailbreak attacks, achieving higher exact match and semantic similarity scores across the ChatGPT-Roles [71], Financial [95], Tomatoes [121], SQuAD2 [113] and SIQA [137] datasets and the GPT-J [168], OPT [188], Falcon [6], LLaMA-2 [160] and Vicuna [26] models.

More than just extracting a prompt, many works focus on manipulating it. In an attempt to challenge the robustness of prompt-aligned language models, [195] present a novel adversarial attack that generates transferable adversarial prompt suffixes. It employs an extension of the Auto-prompt method introduced by Shin et al. [143], a hybrid of greedy and gradient-based search, termed greedy coordinate gradient-based descent (GCG), to automatically identify perturbations that, when appended to a variety of prompts, make the model produce objectionable responses. These adversarial examples are shown to be highly transferable, affecting proprietary LLMs like GPT-3.5 [114], Bard [51], and Claude [9], as well as open source LLMs such as LLaMA-2, Falcon and others.

Guo et al. [55] advance over GCG to formalize the controllable generation of white-box jailbreak attacks on LLMs and establish a connection with controllable text generation. Their work does not rely on the discrete token-level optimization of GCG. Instead, it adapts an energy-based constrained decoding algorithm using Langevin Dynamics introduced by Welling and Teh [171], termed COLD, to perform efficient gradient-based sampling in the continuous logit space, before decoding them back into discrete texts. This attack integrates control parameters—such as fluency, stealth, sentiment, and left-right coherence—to generate adversarial attacks in a unified manner. It supports both conventional fluent suffix attacks and novel scenarios, including adversarial paraphrasing and position-constrained stealthy insertions. Experiments on Llama-2, Mistral, Vicuna, Guanaco, GPT-3.5, and GPT-4 demonstrate the framework's high success rate, robust controllability, and effective transferability.

Ren et al. [132] introduce ActorAttack, a multi-turn jailbreak method that uses self-discovered clues to guide LLMs toward producing harmful outputs. Rooted in actor-network theory, the approach builds a network of semantically linked "actors"—both human and non-human—as diverse attack clues related to a harmful target. In the pre-attack phase, the method samples these clues to obtain potential triggers. Then, using a self-talk mechanism, ActorAttack infers an attack chain that guides the generation of a multi-turn query set. Finally, dynamic modification refines this path based on victim responses, leveraging a GPT-4-based judge. The harmfulness of models is evaluated on HarmBench [96]. Overall, ActorAttack automates the discovery of diverse multi-turn attack paths, significantly improving success rates, even for GPT-4. Jaech et al. [66] introduce SafeMTData, a dataset for safer LLM alignment. This dataset has been published to facilitate safety alignment training, assisting LLMs in improving their resilience against sophisticated multi-turn jailbreak-style prompt attacks.

Recently, Li et al. [90] attempted to tackle the poor transferability across models and high computational cost caused by sequential token replacement. To address these, they introduced TF-Attack, a black-box attack that uses an external LLM (Llama2-7B in the experiments) to assess token importance and group tokens into "Importance Levels". This grouping allows for parallel substitutions, significantly reducing the attack time. In addition, the framework proposes the Multi-Disturb and Dynamic-Disturb techniques to increase both the efficiency and transferability of the adversarial examples. Experimental results on six benchmarks (Yelp Polarity and AG News [190], SNLI [19], IMDB [63], MR [121] and MNLI [172]) show that TF-Attack outperforms previous methods achieving over a 10× speedup while maintaining language fluency and significantly impairing the performance of various victim models.

More recently, in early 2025, [89] presents a method that exploits vulnerabilities of LLM-powered agents, using their external integrations—such as memory systems, web access, and API interactions—to conduct simple attacks. The authors first categorize these vulnerabilities into a taxonomy and then demonstrate a series of practical attacks on Anthropic's Computer Use web agent and MultiOn that can, for example, leak private data such as credit card numbers, download malicious files, and send phishing emails, all without requiring any specialized ML knowledge. These attacks expose critical security risks in commercial systems that could lead to massive privacy breaches and financial losses in real-world deployments.

However, attacks on LLMs are not restricted to jailbreak. ICLAttack introduced by Zhao et al. [192] is a backdoor method specifically designed for LLM in-context learning used by Dong et al. [40] (ICL). ICLAttack is achieved through two strategies, namely, poisoning demonstration examples and poisoning demonstration prompts. In the former, sentence-level triggers are inserted into a subset of demonstration examples while preserving their correct labels, so that the attack remains stealthy. In the latter, the method replaces standard prompt templates with adversarial ones that serve as triggers, enabling the backdoor to be activated even when the user's query is unaltered. The core idea of ICLAttack is to exploit the analogical reasoning capability of ICL, whereby the model learns to associate the inserted trigger with a target label. Once the poisoned demonstration context is constructed, any user query that either contains the trigger (in the case of poisoned examples) or is processed with the malicious prompt (in the case of poisoned prompts) leads the model to output the attacker's predefined target label. Experiments conducted on multiple text classification datasets (such as SST-2 [148], OLID [183], and AG News [190]) and across various LLM architectures (including OPT [188], GPT-NEO [18], GPT-J [168], and Falcon [6]) demonstrate that ICLAttack achieves an average attack success rate exceeding 95% while only minimally affecting clean accuracy.

In another work Alber et al. [5] discuss the vulnerability of medical LLMs to data-poisoning attacks by simulating corruption of The Pile [49], a large training dataset, with minute fractions of AI- rs train multi-billion-parameter models on the poisoned datasets and demonstrate that even a 0.001% replacement of training tokens significantly increases the likelihood of generating malicious medical outputs. To address these risks, they suggest a defense technique utilizing the hierarchical nature of biological knowledge graphs to evaluate and filter LLM outputs, achieving high precision and recall in identifying misinformation while successfully limiting the impact of data poisoning. They evaluated their method on the LAMBADA [76] and HellaSwag [184] datasets for common-sense language tasks, while for medical tasks, they used MedQA [68], PubMedQA [69], MedMCQA [119] and the MMLU [59] clinical knowledge and professional medicine subsets, using a GPT-3-like LLM.

Attack-in-the-Chain introduced by Liu et al. [91] (AttChain) utilizes chain-of-thought prompting to iteratively generate adversarial examples that boost a target document's ranking in neural retrieval systems. AttChain focuses on exploiting vulnerabilities in information retrieval by dynamically perturbing target documents guided by high-ranking anchor documents. Its approach—filtering anchor documents via a Zipf-based strategy and assigning perturbation budgets based on ranking discrepancies—proves that LLM reasoning can be used for subtle, multi-step adversarial attacks in black-box settings. The experiments are conducted with GPT-3.5 and Llama3 as attackers on the MS MARCO Document Ranking [110] and TREC DL19 [33] datasets.

PentestGPT introduced by Deng et al. [38] is an LLM-based framework that automates penetration testing. Using GPT-3.5, GPT-4, and Bard, it performs real-world security tasks through a structured benchmarking system that covers 13 targets and 182 sub-tasks from HackTheBox and VulnHub. The framework consists of a reasoning module to track progress, a generation module to transform tasks into commands, and a parsing module to process feedback. An active feedback loop ensures human testers validate execution. Evaluations show GPT-4 outperforms other models, demonstrating strong task completion rates but facing challenges with context retention and hallucination. Experiments are performed on PentestPerf, their penetration testing benchmark, to evaluate the performance of penetration testers and automated tools across a wide range of testing targets. While promising for security assessments, the study also demonstrates the risks of LLM misuse in automated cyber-attacks.

More recently, AutoAttacker proposed by Xu et al. [178] presents a system that automates "hands-on-keyboard" cyber-attacks through a modular design incorporating summarization, planning, navigation, and an experience manager. In contrast to PentestGPT, which is not fully automated, as the penetration tester has to act as the proxy between the capture the flag (CTF) environment and the LLM to facilitate their communications, AutoAttacker is designed for executing complex post-breach attacks end-to-end. It breaks down attack tasks into manageable subtasks and reuses successful actions through retrieval-augmented techniques, showing the capacity of LLMs to generate precise, context-aware attack commands with minimal human intervention. AutoAttacker is evaluated on custom benchmark, broader than PentestPerf which focuses on the CTF setup, and GPT-4 is chosen as the attacker model due to its higher performance compared to GPT-3 and Llama2.

Lastly, the application of Large Language Models to distributed denial-of-service (DDoS) Attack Detection first demonstrated by Guastalla et al. [53] adopts LLMs in a defensive role, utilizing few-shot and fine-tuning techniques to accurately detect DDoS attacks in IoT networks. This work is different from the offensive objectives of the above papers. While they demonstrate how LLMs can be co-opted to automate cyber-attacks, the DDoS detection study shows that, when properly prompted, LLMs can serve as effective defense mechanisms by classifying and explaining potential threats with high accuracy. Experiments confirm the claims on the CICIDS 2017 [141] and Urban IoT [58] datasets, showing that LLMs with few-shot learning outperform fully supervised multi-layer perceptrons (MLPs).

6. Defense mechanisms and mitigations

Modern AI systems must deal with a growing number of security threats, including poisoning and evasion attacks, as well as LLM-specific vulnerabilities, inference attacks, model extraction, and model inversion. The MITRE ATLAS framework provides a systematic collection of AI adversarial techniques, identifying specific methods that attackers can employ to harm AI models and their associated data. By mapping real-world threats to their associated ATLAS techniques, we can determine particular defenses and mitigation strategies. This approach enables clear knowledge of which defensive measures are most effective against certain threats, and serves as a strong foundation for developing powerful AI systems.

This section examines each defense approach, including its aim, methodology, and particular vulnerabilities addressed. Collectively, these mitigations provide a realistic and practical paradigm for strengthening the security, integrity, and robustness of AI systems throughout the development and deployment process. Table 1 provides an in-depth analysis of each attack, including its ATLAS technique and related defensive and mitigation strategies. Serving as a thorough reference for professionals as well as researchers to better understand the connections between adversarial techniques and defense mechanisms. Furthermore, the Table 1 shows that certain defenses can be utilized against a variety of attack strategies. Key defenses that can be applied to all of the examined attack categories are described below:

Verify AI Artifacts: In order to ensure that the file has not been modified by a malicious party, it is necessary to validate the cryptographic checksum of every single AI artifact.

AI Bill of Materials: An AI Bill of Materials (AI BOM) lists all of the artifacts and resources utilized to develop the AI. The AI BOM can assist in reducing supply chain risks and enabling rapid adaptation to detected vulnerabilities. This might involve preserving dataset provenance, or a complete history of datasets used in AI applications. The history might contain information about the dataset's origin as well as a detailed record of any changes.

Limit Model Artifact Release: Limit the public distribution of technical project-specific information such as data, algorithms, model structures, and model checkpoints that are or will be utilized in production.

Control Access to AI Models and Data at Rest: Establish access restrictions for internal model registries and restrict internal access to production models. Only approved users should have access to training data.

Sanitize Training Data: Detect, eliminate, or remediate poisoned data from training. Sanitizing training data before model training is recommended, as well as on a regular basis for active learning models. Implement a filter to restrict the amount of training data that is consumed. Create a content policy to prevent the use of inappropriate content, such as explicit or offensive language.

Maintain AI Dataset Provenance: Maintain a precise history of datasets utilized by AI applications. The history should include information related to the dataset's origins as well as a detailed record of any changes.

Generative AI Guardrails: Guardrails are safety restrictions that are added between a generative AI model and its outcome shared with the user to avoid unwanted inputs and outputs. Guardrails can include validators like filters, rule-based logic, or regular expressions, as well as AI-based techniques like classifiers and the use of LLMs or named entity recognition (NER) to assess the safety of the prompt or answer. Domain-specific techniques can be used to mitigate risks in a range of fields, including brand reputation, SQL injection attacks, potential data leaks, misinformation, etiquette, code vulnerabilities, and jailbreak attempts.

Model Hardening: Adversarial training or network distillation are two strategies for making AI models resilient to adversarial inputs.

Use Ensemble Methods: To improve resilience against adversarial inputs, use an ensemble of models for inference. Certain models or model families may be successfully evaded by certain attacks, whereas others may not be.

Input Restoration: All inference data should be preprocessed to eliminate or reverse potential adversarial perturbations.

Adversarial Input Detection: Detect and prevent adversarial inputs or unusual queries that differ from known benign behaviors, display behavior patterns observed in past attacks or originate from potentially hostile IP addresses. Incorporate adversarial detection techniques into the AI system prior to the AI model.

Use Multi-Modal Sensors: Incorporate multiple sensors to combine different views and modalities to prevent a single point of failure that is vulnerable to a physical attacks.

AI Model Distribution Methods: Deploying AI models on edge devices might enhance the system's attack surface. Consider providing models on the cloud to restrict the adversary's access to the model. Consider cloud computing features to avoid gray-box attacks, which occur when an attacker has access to model preparation procedures.

Passive AI Output Obfuscation: Reducing the accuracy of model outputs presented to the end user can limit an adversary's capacity to gather knowledge about the model and improve attacks against it.

Restrict Number of AI Model Queries: Limit the quantity and frequency of requests a user can make.

Generative AI Guidelines: Guidelines are safety restrictions that are placed between user-supplied input and a generative AI model to help guide the model to create desired outputs while preventing undesirable outcomes. Guidelines can be used as instructions attached to all user prompts or as part of the system prompt. They can describe the system's goal(s), role, and voice, as well as establish its safety and security requirements.

Generative AI Model Alignment: It is essential to employ techniques that improve model alignment with safety, security, and content requirements while training or optimizing a generative AI model. The fine-tuning process has the potential to remove built-in safety mechanisms in a generative AI model, but techniques such as Reinforcement Learning from Human or AI Feedback, Supervised Fine-Tuning, and Targeted Safety Context Distillation can improve the model's safety and alignment.

AI Telemetry Logging: Log the inputs as well as outputs from deployed AI models. Monitoring logs can assist in detecting security issues and mitigating their effects. Additionally, enabling logging could discourage adversaries who wish to remain undiscovered from using AI resources.

User Training: Teach AI model developers about secure coding methods and AI vulnerabilities.

Restrict Library Loading: Configure proper library loading mechanisms within the operating system and applications to prevent the loading of untrusted code. Investigate potentially vulnerable software. File formats used for storing AI models, such as pickle files, may include exploits that allow malicious libraries to be loaded.

Code Signing: To prevent untrusted code from running, enforce binary and application integrity via digital signature verification. Adversaries have the ability to embed harmful malware in AI software or models. Code signing enforcement can help to keep the AI supply chain secure and prevent malicious code from executing.

Vulnerability Scanning: Vulnerability scanning can be utilized to identify potentially exploitable software vulnerabilities and resolve them. File formats, such as pickle files, which are often utilized for storing AI models, might include bugs that allow arbitrary code execution. These files should be inspected for potentially dangerous calls that might be used to run code, create new processes, or enable networking. Adversaries may encode dangerous code in corrupt model files, therefore scanners must be able to deal with models that can't be completely de-serialized. Model artifacts and downstream products should be inspected for known vulnerabilities.

Encrypt Sensitive Information: Encrypt sensitive data, such as AI models, to prevent unauthorized access.

Limit Public Release of Information: Limit the amount of technical information about an organization's AI stack that is made available to the public. Adversaries can utilize technical understanding of how AI works to target and customize attacks on the target system. Consider restricting the sharing of organizational information, such as geographical locations, researcher names, and department structures, from which technical details including AI methods, model architectures, or datasets might be extracted.

7. Discussion and future directions

A closer look at the MITRE ATLAS case studies shows that adversarial attacks often build on each other. Rather than using just one technique, attackers rely on multi-stage processes where each step sets up the next. For example, an adversary might start by mounting a black-box adversarial attack on an ML model offered as-a-service. This allows them to generate samples near the model decision boundary, making it easier and quicker to replicate the model behavior. With the proxy model in hand, attackers can then craft white-box adversarial examples to avoid detection with very few API calls. Every stage in this chain magnifies the damage of the one before it, and combining tactics also helps attackers dodge layered security measures. While issues revolving around the trustworthiness of AI models are thoroughly considered on an individual basis, i.e., “model stealing” or “model evasion”, the combination of attacks is not jointly discussed. Current defenses also tend to address single-attack scenarios. While MITRE ATLAS reveals many such incidents throughout the case studies, very little work has been done on compound or sequential attack strategies and how to defend when multiple weak points are exploited.

Recently, XAI has been incorporated into many AI suites to enhance model transparency. For instance, platforms such as Google Cloud’s Explainable AI, IBM Watson, and AWS SageMaker Clarify offer explanations along with their predictive services. However, XAI is not yet represented in the MITRE ATLAS framework. While it facilitates the use of AI by circumventing its opaque nature, XAI also introduces risks by exposing critical information about the model inner workings. Explanations can enable adversaries to better understand a model’s behavior. Consequently, it requires fewer queries to replicate or manipulate the model when explanations are provided. While some discussion does exist by Spartalis et al. [150], it is still relatively understudied. On the other hand, adversaries may also target the explanations themselves. Baniecki and Biecek [13] have highlighted scenarios where explanations are adversarially manipulated, while work by Artelt et al. [10] discusses how poisoning attacks can change explanations without affecting model predictions. Therefore, the requirement for balance between transparency and security in AI systems is becoming more important. MITRE ATLAS does not currently cover XAI-specific threats, such as attacks on explanations or model transparency, indicating a gap that must be filled.

With the rise of LLMs, Reinforcement Learning (RL) has regained popularity. Specifically, Reinforcement Learning from Human Feedback (RLHF) by Ouyang et al. [117] and Group Relative Policy Optimization (GRPO) by Shao et al. [140] are used to instruct LLMs, following their supervised training. However, RL is vulnerable to poisoning attacks, as adversaries can manipulate rewards, environments, or training data. These attacks cause RL agents to adopt suboptimal policies or act maliciously when triggered. Methods such as reward poisoning, adversarial environment manipulation, and backdoor attacks pose significant risks. Additionally, attacking RL agents is more dangerous than ever due to the popularity of deploying agentic workflows. Therein, the attack landscape expands with the number of agents involved. Currently, the multi-agent attack surface (one AI exploiting another) is largely underexplored. It is significant for any domain where AI systems cooperate or compete: from robotics (drone fleets) to finance (automated trading agents interacting) to cybersecurity (automated defenders vs. attackers). Traditional adversarial ML focuses on single-model vulnerabilities, so expanding to multi-agent contexts requires more research. MITRE ATLAS currently provides only a limited coverage of RL-specific threats and does not describe multi-agent exploitation patterns, coordination-based threats, or cross-agent influence pathways. Addressing these gaps is critical, as RLHF-driven and agentic systems rapidly guide the operational behavior of large AI deployments.

Another unexplored area revolves around multi-modal models as shown by Baltrušaitis et al. [12], where text, image, audio, video and other modalities, are processed simultaneously by the same model. Despite the enhanced capabilities they bring, owing to stronger signals from multiple data sources, multi-modal ML raises new security issues regarding their vulnerability to attacks. While there has been some work in the field by Dou et al. [41], adversarial attacks are heavily understudied in multi-modal models. For instance, how does an adversary effectively poison one modality to compromise the overall model, and can perturbations in a less influential channel amplify vulnerabilities in others? Similarly, when attempting model extraction or stealing attacks, is one modality inherently more exploitable than another? Furthermore, can cross-modal interactions be used to perform inversion attacks that reconstruct sensitive training data from partial inputs, and do these interactions offer resilience or fragility during inference under adversarial conditions? In real-world applications, such as in an AI-powered content filter that checks both text and images, a cross-modal adversary could evade detection by distributing the malicious cue across modalities. Current defenses also tend to address unimodal attack scenarios. The current MITRE ATLAS taxonomy does not comprehensively include modality-specific attack vectors, cross-modal transferability, or multimodal inversion issues, despite their growing importance in foundation models. Extending the paradigm to include multi-modal threat categories would allow for more comprehensive threat modeling in current multisensor systems.

Another field gaining popularity is neurosymbolic AI, which combines neural networks with symbolic reasoning or logic-based components. This hybrid approach improves interpretability and reasoning, but it also introduces new vulnerabilities for both the neural and symbolic domains. An important question is whether adding symbolic structure makes the system more robust or more sensitive to adversarial manipulation. The answer so far is not clear. [134] suggests that certain neurosymbolic architectures can be more adversarially robust than purely neural ones, for example if the symbolic module provides constraints that limit the neural network susceptibility to nonsensical perturbations. However, if symbolic rules are too simplistic (an “interpretable shortcut”), an adversary can exploit that to break the system, even if the neural part is robust. This area remains underexplored—the attack surface includes manipulating the neural network’s inputs or the knowledge base/rules that the symbolic component uses. For instance, an attacker might add a few fake facts to a knowledge graph that a neurosymbolic system consults, leading the AI to draw dangerously wrong conclusions (a form of symbolic poisoning). Such hybrid architectures are not fully captured in MITRE ATLAS, indicating a structural gap for capturing symbolic poisoning, logic-level adversarial manipulation, or hybrid neural–symbolic exploit chains. As neurosymbolic systems expand, the taxonomy must evolve accordingly.

Another critical research avenue that remains largely unexplored is security for continual learning. While traditional ML models have a fixed behavior after training, continual learning systems update their knowledge or adapt over time based on new data or feedback. Examples include reinforcement learning systems that keep training in deployment, or LLM-based agents that refine their responses via user feedback. While this adaptability is powerful, it also means the model behavior is a changing attack surface. An attacker might gradually influence a self-learning system off course—a form of continual poisoning. As per Cisco’s AI security report [30], when AI applications continue to learn from new data, “new vulnerabilities and emergent behavior can appear after deployment, unlike traditional software that does not change unless you change it”. This requires continuous monitoring and periodic robustness re-evaluation. Most research still treats defense as a one-time process, while the community needs online methods and mechanisms to shut down critical components of a model even as other parts learn, in order to prevent drifting into a compromised state. MITRE ATLAS does not completely capture such hybrid architectures, revealing a structural gap in the detection of symbolic poisoning, logic-level adversarial manipulation, or hybrid neural-symbolic attack chains. The taxonomy of neurosymbolic systems must grow in parallel with their expansion.

7.1. Framework limitations and application security

To address the challenge of standardization in AI security, it is essential to situate MITRE ATLAS within the current tri-polar landscape of defense frameworks. Governance standards such as the NIST AI RMF [105] and ISO/IEC 42,001 [65] provide the high-level “why” and “what” of organizational risk management, whereas application security standards such as the OWASP Top 10 for LLM Applications [118] offer the developer-focused “where” in terms of vulnerabilities. MITRE ATLAS fills the unique gap of “how” in this ecosystem, functioning not as a compliance checklist, but instead as a dynamic threat framework derived from MITRE ATT&CK.

While MITRE ATLAS presents a thorough taxonomy of adversarial tactics, it is critical to recognize its limitations as a “living knowledge base”. MITRE ATLAS depends primarily on real-world case studies as well as real red-teaming scenarios to populate its matrix. This empirical approach creates a codification lag, in which theoretical vulnerabilities reported in academic literature are not included in the framework until they’ve been operationalized in the wild. As a result, the framework may underestimate novel threats from emerging fields like neurosymbolic AI or multi-modal systems, where public events are rare.

Furthermore, the dependence upon voluntary incident reporting leads to reporting bias. High-visibility attacks, such as chatbot manipulation, are widely reported; however silent failures, such as model extraction or data leakage, remain unreported because of intellectual property concerns or a lack of detection. As a result, application-level guidelines, such as the OWASP Top 10 for LLM Applications, are beneficial since they emphasize deployment-specific concerns. While ATLAS concentrates on the adversary’s perspective TTPs, OWASP tackles the developer’s perspective by documenting significant application weaknesses such as insecure output handling and supply chain vulnerabilities. Thus, combining ATLAS and OWASP provides a more comprehensive knowledge of LLM and AI application security, with every framework providing separate and complimentary insights about threat behaviors and system-level vulnerabilities.

Overall, these findings highlight both the importance of MITRE ATLAS as a framework as well as the areas where the community can expand and operationalize it. Future investigations should attempt to address gaps in coverage, increase granularity and benchmarking, and integrate future attack classes (XAI, multi-agent, multi-modal, neurosymbolic, continuous learning). Finally, additional research should include mappings to measurable defense metrics, as there are no standardized metrics or Key Performance Indicators (KPIs) to assess how well a defense addresses MITRE ATLAS techniques, transforming MITRE ATLAS into a comprehensive and empirically based resource for adversarial machine learning research and secure AI engineering.

7.2. Operationalizing ATLAS for enterprise defense and regulatory compliance

MITRE ATLAS is based on the industry-standard MITRE ATT&CK architecture, allowing enterprises to efficiently integrate AI threat intelligence into their existing Security Operations Centers (SOCs). Security analysts who are already familiar with TTP-based approaches for traditional IT can utilize ATLAS to extend their threat detection and incident response playbooks to AI systems. Particularly, the defensive mechanisms described in Section 6 serve as a foundation for these playbooks. Organizations can shift from reactive patching towards proactive hardening of their ML pipelines by mapping known adversarial behaviors to particular mitigations(e.g., adversarial training, input sanitization). This is particularly vital in critical infrastructures such as energy or healthcare grids which employ predictive maintenance or diagnostic AI, since a harmful attack can result in physical disruptions.

Beyond operational security, ATLAS provides a systematic approach to regulatory compliance. Emerging frameworks, such as the EU AI Act [43], mandate manufacturers of “high-risk” AI systems to demonstrate robustness against adversarial threats and guarantee cybersecurity resilience. Similarly, standards such as ISO/IEC 42,001 [65] and the NIST AI Risk Management Framework (AI RMF) [105] place particular emphasis on the “measure” and “manage” functions for adversarial attacks. Our mapping operationalizes these criteria by identifying potential attack vectors employing the ATLAS taxonomy. By applying the defenses outlined in Section 6, practitioners can systematically document their security posture during compliance evaluations. Consequently, rather than simply serving as a descriptive attack matrix, ATLAS is an important auditing tool for demonstrating due diligence in an increasingly regulated industry.

8. Conclusions

In conclusion, this survey has offered an in-depth exploration of adversarial attacks on AI systems through the lens of the MITRE ATLAS framework. In Section 3, we outlined the fundamental tactics, objectives, and techniques that adversaries use, supporting our discussion with real-life case studies that demonstrate how these attacks can lead to significant financial losses, erode trust in AI, and damage reputations.

Thereafter, the MITRE ATLAS techniques were categorized according to the literature into six broad areas —Evasion, Poisoning, Model Extraction, Inference, Model Inversion, and LLM Attacks. A total of 63 papers were analyzed in detail providing their categorization, overview, theoretical advances over previous related works, threat models, datasets and experimental results. Our research demonstrated that these threats are not limited to a single domain, but rather span multiple domains and data modalities, extending from traditional CV and NLP to GNNs and RL systems. This review demonstrates that adversarial methods are not isolated tactics, rather, they often interact, allowing attackers to exploit weaknesses at multiple stages of the AI lifecycle. In Section 6, we proposed a systematic mapping of defense mechanisms to ATLAS techniques so that the defensive aspect of this lifecycle could be addressed directly. This paper provides a core paradigm for researchers and practitioners who want to build robust AI systems by linking specific attacks to appropriate mitigations.

Last, we discussed open research avenues, highlighting the need for synergistic approaches that address the multi-stage and interdependent nature of adversarial attacks. We discussed the importance of exploring combined tactics, emerging vulnerabilities introduced by explainable AI and continuous learning frameworks, as well as promising directions in agentic workflows and neurosymbolic AI. Additionally, we identified structural limitations in the current MITRE ATLAS taxonomy and offered specific modifications that could enhance its granularity and coverage of novel and emerging attack vectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Morris II worm: rag-based attack, 2024, https://arxiv.org/abs/2403.02817. Case Study: AML.CS0024.

[2] S.C.V. Ab, IDA2016Challenge. UCI Machine Learning Repository, 2016, https://doi.org/10.24432/C5V60Q

[3] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: a survey, IEEE Access 6 (2018) 14410–14430.

[4] N. Akhtar, A. Mian, N. Kardan, M. Shah, Advances in adversarial attacks and defenses in computer vision: a survey, IEEE Access 9 (2021) 155161–155196.

[5] D.A. Alber, Z. Yang, A. Alyakin, E. Yang, S. Rai, A.A. Valliani, J. Zhang, G.R. Rosenbaum, A.K. Amend-Thomas, D.B. Kurland, et al., Medical large language models are vulnerable to data-poisoning attacks, Nat. Med. (2025) 1–9.

[6] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, É. Goffinet, D. Hesslow, J. Launay, Q. Malartic, et al., The falcon series of open language models, arXiv preprint arXiv:2311.16867, 2023.

[7] M. Andriushchenko, F. Croce, N. Flammarion, 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.

[8] P. Olson, Faces are the next target for fraudsters, Wall St. J. (2021).

[9] Anthropic, The claude 3 model family: opus, sonnet, haiku, Anthropic (2024) https://www.anthropic.com/news/claude-3-family.

[10] A. Artelt, S. Sharma, F. Lecué, B. Hammer, The effect of data poisoning on counterfactual explanations, arXiv preprint arXiv:2402.08290, 2024.

[11] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, arXiv preprint arXiv:1807.00459v3, 2018.

[12] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: a survey and taxonomy, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2018) 423–443.

[13] H. Baniecki, P. Biecek, Adversarial attacks and defenses in explainable artificial intelligence: a survey, Inf. Fusion (2024) 102303.

[14] B. Becker, R. Kohavi, Adult. UCI Machine Learning Repository, 1996, https://doi.org/10.24432/C5XW20

[15] A.N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, arXiv preprint arXiv:1811.12470v4, 2018.

[16] B. Biggio, B. Nelson, P. Laskov, Poisoning attacks against support vector machines, arXiv preprint arXiv:1206.6389v3, 2012.

[17] G. BigQuery, Reddit comments dataset, https://bigquery.cloud.google.com/dataset/fh-bigquery:reddit_comments.

[18] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, et al., GPT-NeoX-20B: An open-source autoregressive language model, arXiv preprint arXiv:2204.06745, 2022.

[19] S.R. Bowman, G. Angeli, C. Potts, C.D. Manning, A large annotated corpus for learning natural language inference, arXiv preprint arXiv:1508.05326, 2015.

[20] W. Brendel, J. Rauber, M. Bethge, Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, arXiv preprint arXiv:1712.04248, 2017.

[21] Brown University Researchers, The dangerous AI open source dilemma (gpt-2 replication), (2019), Wired Case Study: AML.CS0007.

[22] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, F. Tramer, Membership inference attacks from first principles, arXiv preprint arXiv:2112.03570v2, 2021.

[23] N. Carlini, D. Paleka, K. Dvijotham, T. Steinke, J. Hayase, A.F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conmy, I. Yona, E. Wallace, D. Rolnick, F. Tramèr, Stealing part of a production language model, arXiv preprint arXiv:2403.06634v2, 2024.

[24] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 Ieee Symposium on Security and Privacy (Sp), IEEE, 2017, pp. 39–57.

[25] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, Zoo: zeroth order optimization based black-box attacks to deep neural networks without training substitute models, in: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 15–26.

[26] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J.E. Gonzalez, I. Stoica, E.P. Xing, Vicuna: an open-source chatbot impressing gpt-4 with 90%* Chatgpt quality, 2023. https://lmsys.org/blog/2023-03-30-vicuna/.

[27] Y. Choi, Y. Uh, J. Yoo, J.-W. Ha, Stargan V2: diverse image synthesis for multiple domains, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8185–8194.

[28] A. Choquette-Choo, F. Tramer, N. Carlini, N. Papernot, Label-only membership inference attacks, arXiv preprint arXiv:2007.14321v3, 2020.

[29] A.E. Cinà, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B.A. Moser, A. Oprea, B. Biggio, M. Pelillo, F. Roli, Wild patterns reloaded: a survey of machine learning security against training data poisoning, ACM Comput. Surv. 55 (2023) 1–39.

[30] CISCO, AI application security report, https://www.cisco.com/site/us/en/learn/topics/artificial-intelligence/ai-application-security.html#::text=Additionally.

[31] G. Cohen, S. Afshar, J. Tapson, A. Van Schaik, Emnist: extending mnist to handwritten letters, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2921–2926.

[32] International Warfarin Pharmacogenetic Consortium, Estimation of the warfarin dose with clinical and pharmacogenetic data, N. Engl. J. Med. 360 (2009) 753–764, https://doi.org/10.1056/NEJMoa0809329

[33] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, E.M. Voorhees, Overview of the trec 2019 deep learning track, arXiv preprint arXiv:2003.07820, 2020.

[34] F. Croce, M. Hein, Minimally distorted adversarial examples with a fast adaptive boundary attack, in: International Conference on Machine Learning, PMLR, 2020, pp. 2196–2205.

[35] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: International Conference on Machine Learning, PMLR, 2020, pp. 2206–2216.

[36] R. Cyber, Tesla model s and model 3 prove vulnerable to GPS spoofing attacks as autopilot navigation steers Car off road: research from regulus cyber shows vulnerabilities in gnss-dependent autonomous systems, 2019. Press Release.

[37] J. Dai, C. Chen, A backdoor attack against LSTM-based text classification systems, arXiv preprint arXiv:1905.12457v2, 2019.

[38] G. Deng, Y. Liu, V. Mayoral-Vilches, P. Liu, Y. Li, Y. Xu, T. Zhang, Y. Liu, M. Pinzger, S. Rass, PentestGPT: an LLM-empowered automatic penetration testing tool, arXiv preprint arXiv:2308.06782, 2023.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.

[40] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, et al., A survey on in-context learning, arXiv preprint arXiv:2301.00234, 2022.

[41] Z. Dou, X. Hu, H. Yang, Z. Liu, M. Fang, Adversarial attacks to multi-modal models, in: Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis, 2023, pp. 35–46.

[42] S. Edunov, M. Ott, M. Auli, D. Grangier, Understanding back-translation at scale, arXiv preprint arXiv:1808.09381, 2018.

[43] European Parliament and Council of the European Union, Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (artificial intelligence act), Off. J. Eur. Union L (2024) http://data.europa.eu/eli/reg/2024/1689/oj.

[44] Eyepacs, Diabetic retinopathy detection, 2018, https://www.kaggle.com/c/diabetic-retinopathy-detection (Accessed: 8 November 2018).

[45] H. Fang, Y. Qiu, H. Yu, W. Yu, J. Kong, B. Chong, B. Chen, X. Wang, S.-T. Xia, K. Xu, Privacy leakage on DNNs: a survey of model inversion attacks and defenses, arXiv preprint arXiv:2402.04013, 2024.

[46] R.A. Fisher, Iris. UCI Machine Learning Repository, 1936, https://doi.org/10.24432/C56C76

[47] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, T. Ristenpart, Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing, in: Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14), 2014, pp. 17–-32.

[48] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS), ACM, 2015, pp. 1322–1333, https://doi.org/10.1145/2810103.2813677

[49] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al., The pile: an 800GB dataset of diverse text for language modeling, arXiv preprint arXiv:2101.00027, 2020.

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

[51] Google, Bard: google's generative AI chatbot, 2023. https://bard.google.com.

[52] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset. Available at: 2007. http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

[53] M. Guastalla, Y. Li, A. Hekmati, B. Krishnamachari, Application of large language models to ddos attack detection, in: International Conference on Security and Privacy in Cyber-Physical Systems and Smart Vehicles, Springer, 2023, pp. 83–99.

[54] C. Guo, J. Gardner, Y. You, A.G. Wilson, K. Weinberger, Simple black-box adversarial attacks, in: International Conference on Machine Learning, PMLR, 2019, pp. 2484–2493.

[55] X. Guo, F. Yu, H. Zhang, L. Qin, B. Hu, Cold-attack: jailbreaking LLMs with stealthiness and controllability, arXiv preprint arXiv:2402.08679, 2024.

[56] I. Guyon, Madelon. UCI Machine Learning Repository, 2004, https://doi.org/10.24432/C5602H

[57] G. Han, J. Choi, H. Lee, J. Kim, Reinforcement learning-based black-box model inversion attacks, arXiv preprint arXiv:2304.04625v1, 2023.

[58] A. Hekmati, E. Grippo, B. Krishnamachari, Dataset: large-scale urban iot activity data for DDoS attack emulation, arXiv preprint arXiv:2110.01842, 2021.

[59] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, J. Steinhardt, Measuring massive multitask language understanding, arXiv preprint arXiv:2009.03300, 2020.

[60] H. Hofmann, Statlog (german credit data). UCI Machine Learning Repository, 1994, https://doi.org/10.24432/C5NC77

[61] M. Hopkins, E. Reeber, G. Forman, J. Suermondt, Spambase. UCI Machine Learning Repository, 1999, https://doi.org/10.24432/C53G6X

[62] A. Ilyas, L. Engstrom, A. Athalye, J. Lin, Black-box adversarial attacks with limited queries and information, in: International Conference on Machine Learning, PMLR, 2018, pp. 2137–2146.

[63] IMDb, IMDb Non-Commercial datasets, 2023. https://developer.imdb.com/non-commercial-datasets/.

[64] Incident Database, Microsoft's tay chatbot poisoning, 2016. https://incidentdatabase.ai/cite/6. Case Study: AML.CS0009.

[65] International Organization for Standardization, Iso/iec 42001:2023 artificial intelligence—management system, 2023. https://www.iso.org/standard/81230.html. standard.

[66] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al., OpenAI O1 system card, arXiv preprint arXiv:2412.16720, 2024.

[67] M. Jagielski, N. Carlini, A. Kurakin, N. Papernot, High accuracy and high fidelity extraction of neural networks, arXiv preprint arXiv:1909.01838v2, 2019.

[68] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, P. Szolovits, What disease does this patient have? A large-scale open domain question answering dataset from medical exams, Appl. Sci. 11 (2021) 6421.

[69] Q. Jin, B. Dhingra, Z. Liu, W.W. Cohen, X. Lu, PubMedQA: a dataset for biomedical research question answering, arXiv preprint arXiv:1909.06146, 2019.

[70] JKraak, Bitcoin price dataset, 2023. https://www.kaggle.com/datasets/jkraak/bitcoin-price-dataset.

[71] W. Jones, ChatGPT roles, 2023, https://huggingface.co/datasets/WynterJones/chatgpt-roles (Accessed: 3 April 2024).

[72] Kaggle, Acquire valued shoppers challenge, 2015, https://www.kaggle.com/competitions/acquire-valued-shoppers-challenge (Kaggle competition. Accessed: 24 February 2025).

[73] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, T. Aila, Training generative adversarial networks with limited data, in: Conference on Neural Information Processing Systems (NeurIPS), 2020.

[74] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4401–4410.

[75] Kaspersky ML Research Team, Confusing antimalware neural networks, 2021. https://securelist.com/how-to-confuse-antimalware-neural-networks-adversarial-attacks-and-protection/102949/. Case Study: AML.CS0014.

[76] M. Kazemi, N. Kim, D. Bhatia, X. Xu, D. Ramachandran, LAMBADA: backward chaining for automated reasoning in natural language, arXiv preprint arXiv:2212.13894, 2022.

[77] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel dataset for fine-grained image categorization, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, 2011.

[78] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE 2007-01, Keele University and Durham University Joint Report, 2007.

[79] P.W. Koh, P. Liang, Understanding black-box predictions via influence functions, arXiv preprint arXiv:1703.04730v3, 2017.

[80] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, technical Report, University of Toronto, 2009, https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf.

[81] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb51: a large video database for human motion recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2556–2563.

[82] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, arXiv preprint arXiv:1611.01236, 2016.

[83] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: Artificial Intelligence Safety and Security, Chapman and Hall/CRC, 2018, pp. 99–112.

[84] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al., The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale, arXiv preprint 5, 8, 11 arXiv:1811.00982, 2018.

[85] lakshmi25npathi, Imdb dataset of 50k movie reviews, https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews/data.

[86] Lasso Security, AI package hallucinations, 2024. https://www.lasso.security/blog/ai-package-hallucinations. Case Study: AML.CS0022.

[87] Q.V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, A.Y. Ng, Building high-level features using large scale unsupervised learning, arXiv preprint arXiv:1112.6209v5, 2011.

[88] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.

[89] A. Li, Y. Zhou, V.C. Raghuram, T. Goldstein, M. Goldblum, Commercial LLM agents are already vulnerable to simple yet dangerous attacks, arXiv preprint arXiv:2502.08586, 2025.

[90] Z. Li, K. Chen, L. Liu, X. Bai, M. Yang, Y. Xiang, M. Zhang, Tf-attack: transferable and fast adversarial attacks on large language models, Knowl.-Based Syst. (2025) 113117.

[91] Y.-A. Liu, R. Zhang, J. Guo, M. de Rijke, Y. Fan, X. Cheng, Attack-in-the-chain: Bootstrapping large language models for attacks against black-box neural ranking models, arXiv preprint arXiv:2412.18770, 2024.

[92] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.

[93] X. Luo, Y. Jiang, X. Xiao, Feature inference attack on shapley values, arXiv preprint arXiv:2407.11359v1, 2024.

[94] A. Madry, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083, 2017.

[95] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, P. Takala, Good debt or bad debt: detecting semantic orientations in economic texts, J. Assoc. Inf. Sci. Technol. 65 (2014) 782–796.

[96] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li, et al., HarmBench: a standardized evaluation framework for automated red teaming and robust refusal, arXiv preprint arXiv:2402.04249, 2024.

[97] C. Meek, B. Thiesson, D. Heckerman, US census data (1990). UCI Machine Learning Repository. 2001), https://doi.org/10.24432/C5VP42

[98] S. Merity, C. Xiong, J. Bradbury, R. Socher, Pointer sentinel mixture models, 2016.

[99] V. Metsis, I. Androutsopoulos, G. Paliouras, SPAM filtering with naive bayes–which naive bayes? in: Proceedings of the Conference on Email and Anti-Spam (CEAS), 2006, pp. 28–69.

[100] S. Milli, L. Schmidt, A.D. Dragan, M. Hardt, Model reconstruction from model explanations, arXiv preprint arXiv:1807.05185v1, 2018.

[101] MITRE Corporation, Mitre atlas: adversarial threat landscape for artificial-intelligence systems, 2021. https://atlas.mitre.org/.

[102] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.

[103] S. Moro, P. Rita, P. Cortez, Bank marketing. UCI Machine Learning Repository. 2014, https://doi.org/10.24432/C5K306

[104] C. Morris, N.M. Kriege, F. Bause, K. Kersting, P. Mutzel, M. Neumann, Tudataset: a collection of benchmark datasets for learning with graphs, 2020.

[105] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), Technical Report NIST AI 100-1, Department of Commerce., U.S., 2023, https://doi.org/10.6028/NIST.AI.100-1

[106] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011, pp. 5.

[107] H. Ng, S. Winkler, A data-driven approach to cleaning large face datasets, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2014, pp. 343–347.

[108] N. Ng, K. Yee, A. Baevski, M. Ott, M. Auli, S. Edunov, Facebook fair's wmt19 news translation task submission, arXiv preprint arXiv:1907.06616, 2019.

[109] N.B. Nguyen, K. Chandrasegaran, M. Abdollahzadeh, N.M. Cheung, Rethinking model inversion attacks against deep neural networks, arXiv preprint arXiv:2304.01669v2, 2023.

[110] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, MS marco: a human-generated machine reading comprehension dataset, 2016.

[111] Z. Niu, M. Zhou, L. Wang, X. Gao, G. Hua, Ordinal regression with multiple output CNN for age estimation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4920–4928, https://doi.org/10.1109/CVPR.2016.532

[112] S.J. Oh, M. Augustin, B. Schiele, M. Fritz, Towards reverse-engineering black-box neural networks, arXiv preprint arXiv:1711.01768v3, 2017.

[113] D. Oliynyk, R. Mayer, A. Rauber, I know what you trained last summer: a survey on stealing machine learning models and defences, ACM Comput. Surv. 55 (2023) 1–41.

[114] OpenAI, GPT-3.5: openai's advanced language model, 2023. https://openai.com/research/gpt-3-5.

[115] T. Orekondy, B. Schiele, M. Fritz, Knockoff Nets: stealing functionality of black-box models, arXiv preprint arXiv:1812.02766v1, 2018.

[116] M. Ott, S. Edunov, D. Grangier, M. Auli, Scaling neural machine translation, arXiv preprint arXiv:1806.00187, 2018.

[117] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Adv. Neural Inf. Process. Syst. 35 (2022) 27730–27744.

[118] OWASP Foundation, Owasp top 10 for large language model applications, 2024, https://www.owasp.org/www-project-top-10-for-large-language-model-applications/ (accessed: 12 February 2025).

[119] A. Pal, L.K. Umapathi, M. Sankarasubbu, MedMCQA: a large-scale multi-subject multi-choice dataset for medical domain question answering, in: Conference on Health, Inference, and Learning, PMLR, 2022, pp. 248–260.

[120] Palo Alto Networks AI Research Team, Botnet domain generation algorithm (dga) detection evasion, 2018. https://faculty.washington.edu/mdecock/papers/byu2018a.pdf. Case Study: AML.CS0001.

[121] B. Pang, L. Lee, Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales, arXiv preprint arXiv:0506075, 2005.

[122] A. Paudice, L. Munoz-Gonzalez, C. Lupu, Label sanitization against label flipping poisoning attacks, arXiv preprint arXiv:1803.00992v2, 2018.

[123] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[124] N. Pinto, Z. Stone, T. Zickler, D.D. Cox, Scaling up biologically-inspired computer vision: a case study in unconstrained face recognition on Facebook, in: Proc. Workshop on Biologically Consistent Vision (In Conjunction with CVPR), 2011.

[125] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, G. Loukas, A taxonomy and survey of attacks against machine learning, Comput. Sci. Rev. 34 (2019) 100199.

[126] P. Putten, Insurance company benchmark (COIL 2000). UCI Machine Learning Repository. 2000, https://doi.org/10.24432/C5630S

[127] PyTorch, Compromised pytorch dependency chain, 2022. https://pytorch.org/blog/compromised-nightly-dependency/. Case Study: AML.CS0015.

[128] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 4–11.

[129] A. Rahmati, S.-M. Moosavi-Dezfooli, P. Frossard, H. Dai, Geoda: a geometric framework for black-box adversarial attacks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8446–8455.

[130] A.S. Rakin, Z. He, D. Fan, TBT: Targeted neural network attack with bit trojan, arXiv preprint arXiv:1909.05193v3, 2019.

[131] E.T. Red, Chatgpt data exfiltration via markdown injection, 2023. https://embracethered.com/blog/posts/2023/chatgpt-webpilot-data-exfil-via-markdown-injection/. Case Study: AML.CS0021.

[132] Q. Ren, H. Li, D. Liu, Z. Xie, X. Lu, Y. Qiao, L. Sha, J. Yan, L. Ma, J. Shao, Derail yourself: multi-turn llm jailbreak attack through self-discovered clues, arXiv preprint arXiv:2410.10700, 2024.

[133] S. Rezaei, X. Liu, On the difficulty of membership inference attacks, arXiv preprint arXiv:2005.13702v3, 2020.

[134] L.E. Richards, J. Yaros, J. Babcock, C. Ly, R. Cosbey, T. Doster, C. Matuszek, On the promise for assurance of differentiable neurosymbolic reasoning paradigms, arXiv preprint arXiv:2502.08932, 2025.

[135] M. Rigaki, S. Garcia, A survey of privacy attacks in machine learning, ACM Comput. Surv. 56 (2023) 1–34.

[136] S. Saeed, S.A. Altamimi, N.A. Alkayyal, E. Alshehri, D.A. Alabbad, Digital transformation and cybersecurity challenges for businesses resilience: issues and recommendations, Sensors 23 (2023) 6666.

[137] M. Sap, H. Rashkin, D. Chen, R. LeBras, Y. Choi, SocialIQA: Commonsense reasoning about social interactions, arXiv preprint arXiv:1904.09728, 2019.

[138] Security O, Shadowray: attack AI workloads actively exploited in the wild, 2024. https://www.oligo.security/blog/shadowray-attack-ai-workloads-actively-exploited-in-the-wild. Case Study: AML.CS0023.

[139] A. Shafahi, W.R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, T. Goldstein, Poison frogs! targeted clean-label poisoning attacks on neural networks, arXiv preprint arXiv:1804.00792v2, 2018.

[140] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y.K. Li, Y. Wu, et al., DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, arXiv preprint arXiv:2402.03300, 2024.

[141] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Intrusion detection evaluation dataset (CIC-IDS2017), in: Proceedings of the of Canadian Institute for Cybersecurity, 2018.

[142] E. Shayegani, M.A. Mamun al, Y. Fu, P. Zaree, Y. Dong, N. Abu-Ghazaleh, Survey of vulnerabilities in large language models revealed by adversarial attacks, arXiv preprint arXiv:2310.10844, 2023.

[143] T. Shin, Y. Razeghi, R.L. Logan IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, arXiv preprint arXiv:2010.15980, 2020.

[144] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, arXiv preprint arXiv:1610.05820v2, 2016.

[145] L. Shumailov, Z. Shumaylov, D. Kazhdan, Y. Zhao, N. Papernot, M.A. Erdogdu, R. Anderson, Manipulating sgd with data ordering attacks, arXiv preprint arXiv:2104.09667v2, 2021.

[146] L. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, R. Anderson, Sponge examples: Energy-latency attacks on neural networks, arXiv preprint arXiv:2006.03463v2, 2020.

[147] Silent Break Security, Proofpoint evasion (proof-pudding), 2020. https://github.com/moohax/Proof-Pudding. Case Study: AML.CS0008.

[148] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng y, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.

[149] K. Soomro, A.R. Zamir, M. Shah, Ucf101: a dataset of 101 human actions classes from videos in the wild, CoRR abs/1212.0402 (2012) http://arxiv.org/abs/1212.0402.

[150] C.N. Spartalis, T. Semertzidis, P. Daras, Balancing XAI with privacy and security considerations, in: European Symposium on Research in Computer Security, Springer, 2023, pp. 111–124.

[151] B. Strack, J.P. DeShazo, C. Gennings, J.L. Olmo, S. Ventura, K.J. Cios, J.N. Clore, Impact of hba1c measurement on hospital readmission rates: analysis of 70, 000 clinical database patient records, BioMed Res. Int. 2014 (2014).

[152] B.E. Strom, A. Applebaum, D.P. Miller, K.C. Nickels, A.G. Pennington, C.B. Thomas, MITRE ATT&CK: Design and Philosophy, Technical Report, The MITRE Corporation, 2018.

[153] L. Struppek, D. Hintersdorf, A. Correia, D. Adler, K. Kersting, Plug & play attacks: Towards robust and flexible model inversion attacks, arXiv preprint arXiv:2201.12179v4, 2022.

[154] L.-F. Stumpp, Achieving code execution in mathgpt via prompt injection, 2021. https://arxiv.org/abs/2103.03874. Case Study: AML.CS0016.

[155] O. Suciu, R. Marginean, D. Kaya III, T. Dumitras, When does machine learning fail? Generalized transferability for evasion and poisoning attacks, in: Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), 2018, pp. 1299–1316.

[156] C. Szegedy, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199, 2013.

[157] TechCrunch, Clearview AI misconfiguration, 2020, Case Study: AML.CS0006.

[158] Texas Department of State Health Services, Texas hospital emergency department research data file (ED-RDF): hospital discharge data public use data file.

[159] J. Torres-Sospedra, M. Raul, A. Martnez-Us, T. Arnau, J. Avariento, UJIIndoorLoc. UCI Machine Learning Repository, 2014, https://doi.org/10.24432/C5MS59

[160] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288, 2023.

[161] F. Tramer, F. Zhang, A. Juels, M.K. Reiter, T. Ristenpart, Stealing machine learning models via prediction apis, arXiv preprint arXiv:1609.02943v2, 2016.

[162] S. Truex, L. Liu, M.E. Gursoy, L. Yu, W. Wei, Demystifying membership inference attacks in machine learning as a service, IEEE Trans. Serv. Comput. 14 (6) (2021) 2073–2089. https://doi.org/10.1109/TSC.2019.2897554

[163] University, Mushroom. UCI Machine Learning Repository. 1981, https://doi.org/10.24432/C5959T

[164] U.S. Department of Justice, New Jersey man sentenced to 6½years in prison for schemes to steal California unemployment benefits, 2023. https://www.justice.gov/usao-edca/pr/new-jersey-man-sentenced-675-years-prison-schemes-steal-california-unemployment. press Release, U.S. Attorney's Office, Eastern District of California.

[165] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Technical Report CNS-TR-2011-001. Available online:, California Institute of Technology, Pasadena, CA, USA, 2011, http://www.vision.caltech.edu/visipedia/CUB-200-2011.html.

[166] K. Wang, Y. Fu, K. Li, A. Khisti, R. Zemel, A. Makhzani, Variational model inversion attacks, arXiv preprint arXiv:2201.10787v1, 2022.

[167] B. Wang, N. Gong, Stealing hyperparameters in machine learning, arXiv preprint arXiv:1802.05351v3, 2018.

[168] B. Wang, A. Komatsuzaki, GPT-j-6b: a 6 billion parameter autoregressive language model, 2021. https://github.com/kingoflolz/mesh-transformer-jax.

[169] X. Wang, Y. Peng, L. Le, Z. Lu, M. Bagheri, R.M. Summers, Chestxray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2097–2106.

[170] H. Wei, H. Tang, X. Jia, Z. Wang, H. Yu, Z. Li, S. Satoh, L. Van Gool, Z. Wang, Physical adversarial attack meets computer vision: a decade survey, IEEE Trans. Pattern Anal. Mach. Intell. 46 (12) (2024) 9797–9817. https://doi.org/10.1109/TPAMI.2024.3430860

[171] M. Welling, Y.W. Teh, Bayesian learning via stochastic gradient langevin dynamics, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), Citeseer, 2011, pp. 681–688.

[172] A. Williams, N. Nangia, S.R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, arXiv preprint arXiv:1704.05426, 2017.

[173] P. Williams, I.K. Dutta, H. Daoud, M. Bayoumi, A survey on security in internet of things with a focus on the impact of emerging technologies, Internet of Things 19 (2022) 100564.

[174] wordsforthewise n.d, Lending club [dataset], https://www.kaggle.com/datasets/wordsforthewise/lending-club.

[175] X. Wu, M. Fredrikson, S. Jha, J.F. Naughton, A methodology for formalizing model-inversion attacks, in: 2016 IEEE 29th Computer Security Foundations Symposium (CSF), 2016, pp. 355–370, https://doi.org/10.1109/CSF.2016.32

[176] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747, 2017.

[177] C. Xie, K. Huang, P.Y. Chen, B. Li, DBA: distributed backdoor attacks against federated learning, in: International Conference on Learning Representations (ICLR) 2020 (2020).

[178] J. Xu, J.W. Stokes, G. McDonald, X. Bai, D. Marshall, S. Wang, A. Swaminathan, Z. Li, Autoattacker: A large language model guided system to implement automatic cyber-attacks, arXiv preprint arXiv:2403.01038, 2024.

[179] C. Yang, Q. Wu, H. Li, Y. Chen, Generative poisoning attack method against neural networks, arXiv preprint arXiv:1703.01340v1, 2017.

[180] D. Yang, Foursquare dataset, 2013. https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

[181] I.-C. Yeh, C.-H. Lien, The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, Expert Syst. Appl. 36 (2009) 2473–2480, https://doi.org/10.1016/j.eswa.2007.12.020

[182] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: attacks and defenses for deep learning, IEEE Trans. Neural Netw. Learn. Syst. 30 (2019) 2805–2824.

[183] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Predicting the type and target of offensive posts in social media, in: Proceedings of NAACL, 2019.

[184] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, Y. Choi, Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830, 2019.

[185] Z. Zenity, Financial transaction hijacking with m365 copilot as an insider, 2024. The Register Case Study: AML.CS0026.

[186] Z. Zhang, M. Chen, M. Backes, Y. Shen, Y. Zhang, Inference attacks against graph neural networks, arXiv preprint arXiv:2110.02631v1, 2021.

[187] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, D. Song, The secret revealer: Generative model-inversion attacks against deep neural networks, arXiv preprint arXiv:1911.07135v2, 2019.

[188] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X.V. Lin, et al., OPT: Open pre-trained transformer language models, arXiv preprint arXiv:2205.01068, 2022.

[189] W.E. Zhang, Q.Z. Sheng, A. Alhazmi, C. Li, Adversarial attacks on deep-learning models in natural language processing: a survey, ACM Trans. Intell. Syst. Technol. 11 (2020) 1–41.

[190] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, Adv. Neural Inf. Process. Syst. 28 (2015).

[191] S. Zhao, X. Ma, M. Zheng, X. Bailey, J. Chen, J. Jiang, Clean-label backdoor attacks on video recognition models, arXiv preprint arXiv:2003.03030v2, 2020.

[192] S. Zhao, M. Jia, L.A. Tuan, F. Pan, J. Wen, Universal vulnerabilities in large language models: Backdoor attacks for in-context learning, arXiv preprint arXiv:2401.05949, 2024.

[193] J. Zhou, Y. Chen, C. Shen, Y. Zhang, Property inference attacks against gans, arXiv preprint arXiv:2111.07608v1, 2021.

[194] F. Zhou, Geographical origin of music. UCI Machine Learning Repository. 2014, https://doi.org/10.24432/C5VK5D

[195] A. Zou, Z. Wang, N. Carlini, M. Nasr, J.Z. Kolter, M. Fredrikson, Universal and transferable adversarial attacks on aligned language models, arXiv preprint arXiv:2307.15043, 2023.

[196] B.U.C. Zuriaga, Medical charges in the USA dataset, 2013. https://bigml.com/user/czuriaga/gallery/dataset/519e25c9925ded7798001542. demographics from Esri at Azure Datamarket; Medical data from cms.gov.

[197] M. Zwitter, M. Soklic, Breast cancer. UCI Machine Learning Repository. 1988, https://doi.org/10.24432/C51P4M