A Survey on Adversarial Machine Learning: Attacks, Defenses, Real-World Applications, and Future Research Directions

Yan Qiao, Nimitha Bangalore Sathyanarayana, Chaowei Shi, Zefeng He, Tao Wang, Tao Hou*

Department of Computer Science and Engineering, University of North Texas, Denton, Texas, USA

Abstract

The rapid proliferation of machine learning (ML) systems across critical domains has heightened concerns about their susceptibility to adversarial threats. This survey offers a comprehensive overview of adversarial machine learning, synthesizing a broad body of research encompassing attack methodologies, defense strategies, and real-world applications. We present a systematic taxonomy of adversarial threats spanning the ML lifecycle, including training-time attacks such as data poisoning and backdoor insertion, as well as inference-time attacks such as evasion, model extraction, and privacy leakage. We examine a wide range of defense mechanisms, including proactive approaches (e.g., adversarial training and input sanitization), detection-based techniques that leverage statistical or behavioral signatures, and reactive strategies such as model patching and ensemble learning. We further discuss recent advances in privacy-preserving machine learning, including differential privacy, federated learning, and secure aggregation. Through real-world case studies in domains such as computer vision, natural language processing, autonomous systems, and healthcare, we highlight persistent vulnerabilities and practical challenges. Finally, we outline critical open problems and promising directions for future research. This work consolidates current understanding and serves as a foundational reference for enhancing the security and robustness of machine learning systems.

Keywords:
Adversarial Machine Learning, Attacks and Defenses, Privacy-Preserving Learning, Security in Machine Learning, Trustworthy Machine Learning, Responsible AI

1. Introduction

In recent years, machine learning (ML) has demonstrated remarkable success across a broad spectrum of domains, ranging from computer vision and natural language processing to healthcare diagnostics and autonomous systems. These advancements have led to the widespread deployment of ML-based systems in both consumer-facing applications and critical infrastructure. As these technologies become more deeply embedded in essential services and high-stakes decision-making processes, ensuring their reliability, robustness, and security has become an urgent concern [1].

Despite their impressive capabilities, ML models, particularly deep neural networks, exhibit unexpected vulnerabilities to small, carefully crafted input perturbations known as *adversarial examples*. These perturbations can induce misclassification or incorrect predictions with high confidence, even though they are often imperceptible to human observers [2]. This phenomenon exposes a fundamental fragility in the design and training of modern ML systems and has motivated the emergence and rapid growth of the research field known as *adversarial machine learning (AML)*.

Adversarial machine learning lies at the intersection of machine learning and security, aiming to understand, exploit, and defend against the inherent weaknesses of ML models [3]. Since the seminal discovery by Szegedy et al. [1], the field has experienced an explosion of research, exploring not only the generation of adversarial inputs but also the broader security implications of deploying ML models in adversarial environments. Adversarial threats are no longer theoretical curiosities; they pose real risks in practical applications. For example, adversarial perturbations can cause autonomous vehicles to misread traffic signs [4], mislead diagnostic tools in medical imaging [5], or allow fraud detection systems in finance to be evaded [6].

Unlike conventional software vulnerabilities, which tend to be discrete and patchable through code fixes, the vulnerabilities in ML systems are often systemic and stem from the underlying statistical nature of model training and inference. The core tension between accuracy and robustness makes it difficult to design models that generalize well on natural data while also resisting adversarial manipulation. Additionally, the high dimensionality of input spaces in deep learning models provides an extensive attack surface. Small perturbations in these spaces can yield disproportionately large effects on model predictions,

*Corresponding Author

Email addresses: yanqiao1@my.unt.edu (Yan Qiao), nimithabangaloresathyanarayana@my.unt.edu (Nimitha Bangalore Sathyanarayana), chaoweishi@my.unt.edu (Chaowei Shi), zefenghe@my.unt.edu (Zefeng He), tao@unt.edu (Tao Wang), tao.hou@unt.edu (Tao Hou)

a property that adversaries routinely exploit.

Further complicating the defenses is the inherently asymmetric nature of adversarial scenarios. Attackers only need to find one successful vector of attack, while defenders must anticipate and mitigate a wide range of potential strategies. This asymmetry is particularly pronounced in black-box settings, where the adversary lacks access to model internals but can still launch successful attacks via transferability or gradient-free methods [7]. The lack of transparency in many production models further hinders effective monitoring, auditing, and verification of security guarantees.

Adversarial machine learning is inherently dynamic and adversarially co-evolving, with new defenses continually prompting stronger attacks and rendering earlier solutions obsolete. At the same time, modern machine learning systems are becoming increasingly complex, incorporating multi-domain, multi-modal, and adaptive architectures that expand the adversarial attack surface. Recent advances in multi-domain generative modeling and multimodal learning for safety-critical applications illustrate this trend [8, 9, 10]. These developments have also motivated robustness-aware approaches that explicitly integrate adversarial considerations into model design and training, including recent adversarially informed learning frameworks and architectures [11, 12]. Moreover, adversarial threats now extend beyond simple evasion to include data poisoning, backdoor insertion, model extraction, and membership inference across different stages of the ML lifecycle.

As research matures, attention is increasingly shifting toward more robust, generalizable, and theoretically grounded defense mechanisms. Techniques such as adversarial training, input preprocessing, model certification, randomized smoothing, and ensemble defenses have shown promise. In parallel, privacy-preserving approaches such as differential privacy, federated learning, and secure aggregation are being adopted to mitigate information leakage and support secure collaborative learning environments [13, 14].

1.1. Scope of the Survey

In this survey, we take a comprehensive look at adversarial machine learning across the full ML pipeline, from attacks to defenses. We focus on three primary dimensions. First, we examine the full attack surface of machine learning systems, such as training-time poisoning attacks and inference-time evasion attacks. We also provide detailed analysis of privacy attacks, model extraction techniques, and other emerging threat vectors that target the unique structural and statistical vulnerabilities of ML models. Second, we systematically categorize defense mechanisms, covering preventive, detection, and reactive techniques, as well as privacy-preserving methods. We place particular emphasis on the theoretical underpinnings and practical viability of these approaches in real-world settings. Third, we explore how adversarial machine learning manifests in different application domains, including variations in attack and defense techniques that emerge from domain-specific characteristics and deployment architectures.

1.2. Survey Methodology

We adopt a systematic methodology to capture the breadth and depth of adversarial machine learning research. Our review encompasses 90 recent papers published in top-tier conferences and journals, including NeurIPS, ICML, ICLR, CCS, USENIX Security, NDSS, and IEEE S&P. From this corpus, we identified the seminal works that form the intellectual backbone of the field. These papers were selected based on their contributions to foundational concepts, novel attack or defense mechanisms, theoretical frameworks for adversarial robustness, demonstrated impact in real-world scenarios, and indicators of promising future research directions.

1.3. Paper Organization

The remainder of this survey is organized to support both comprehensive study and targeted exploration. Section 2 introduces foundational concepts and presents a taxonomy of adversarial machine learning. Sections 3 and 4 examine attacks during training and inference, respectively, detailing their methodologies and implications. Section 5 reviews a wide spectrum of defense strategies, including preventive, detection, reactive, and privacy-preserving techniques. In Section 6, we explore real-world applications and case studies across domains such as healthcare, autonomous systems, and cybersecurity. Section 8 outlines persistent challenges and unresolved issues, while Section 9 highlights promising directions for future research. Finally, Section 10 synthesizes key insights and reflects on the path forward. Each section is crafted to be self-contained while collectively contributing to a comprehensive understanding of adversarial machine learning.

2. Preliminaries and Taxonomy

Adversarial machine learning explores the vulnerabilities of learning algorithms under deliberate manipulation. Before exploring specific attacks and defenses, it is essential to understand the foundational concepts that define the AML landscape. This section provides a brief overview of machine learning principles, introduces the threat models commonly considered in adversarial settings, and classifies adversarial attacks and defenses according to their timing, goals, and knowledge assumptions. The taxonomy presented here serves as a conceptual framework for the remainder of the survey.

2.1. Machine Learning Pipeline

The machine learning pipeline comprises several stages, each of which introduces unique security vulnerabilities. Understanding the pipeline enables a systematic assessment of potential attack vectors [3]. As illustrated in Fig. 1, machine learning systems follow a multi-stage pipeline, from data collection to deployment. Adversarial attacks can target different stages of this pipeline, leading to distinct threat models and security implications.

The pipeline begins with **data collection and preprocessing**, where raw data is gathered and transformed into a format suitable for model training. This stage is particularly susceptible

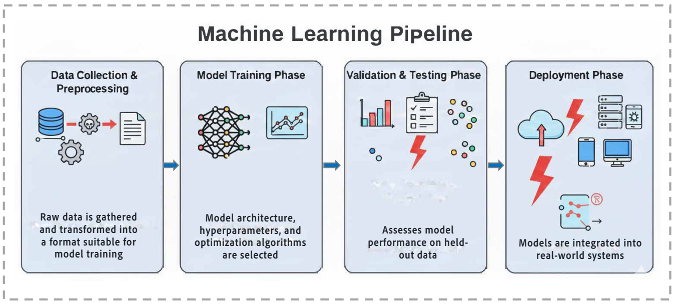
Machine Learning Pipeline
Data Collection & Preprocessing
Raw data is gathered and transformed into a format suitable for model training
Model Training Phase
Model architecture, hyperparameters, and optimization algorithms are selected
Validation & Testing Phase
Assesses model performance on held-out data
Deployment Phase
Models are integrated into real-world systems

Figure 1: Overview of the machine learning pipeline, illustrating the main stages from data collection and preprocessing to model training, validation and testing, and deployment. Each stage may represent a potential attack surface for adversarial manipulation.

to *data poisoning attacks* [15], in which adversaries manipulate training data by injecting malicious samples or modifying existing ones to degrade model performance or introduce targeted behaviors. For instance, slight alterations to training images in computer vision applications can lead to systematic misclassifications.

Next is the **model training phase**, where model architecture, hyperparameters, and optimization algorithms are selected. This stage is vulnerable to attacks such as *backdoor insertions* and *gradient manipulation*. The widespread adoption of deep learning frameworks like PyTorch and TensorFlow, while simplifying training, has inadvertently expanded the attack surface by exposing internal training procedures to manipulation [1].

The **validation and testing phase** assesses model performance on held-out data. Adversaries may construct validation sets that obscure model flaws or manipulate evaluation metrics. Attack samples may appear benign during validation but degrade performance under specific conditions, undermining the model's reliability.

Finally, during the **deployment phase**, models are integrated into real-world systems. This phase raises concerns about model serving infrastructure, exposed APIs, and interaction with external components. Cloud-based deployment increases accessibility but also expands the attack surface through shared infrastructure and network exposure [2].

2.2. Threat Model Framework

Understanding adversarial machine learning requires a clear definition of the adversary's capabilities, goals, and access levels. The threat model framework provides a structured way to characterize these elements, guiding the development and evaluation of both attacks and defenses. In this subsection, we outline the key dimensions of the threat model including the attacker's knowledge of the model (white-box, black-box, or gray-box), and the attack objective (e.g., confidentiality attacks, integrity attacks, or availability attacks). Establishing this framework ensures consistency and clarity in analyzing adversarial behaviors throughout the survey.

2.2.1. Adversary's Knowledge

Threat models are defined by the adversary's level of knowledge:

- **White-box attacks** assume complete access to the model, including architecture, parameters, and training data. These attacks, such as those introduced by Szegedy et al. [1], often employ gradient-based methods to craft highly effective adversarial examples.
- **Gray-box attacks** assume partial knowledge, such as the model architecture or a surrogate trained on a similar dataset.
- **Black-box attacks** assume only input-output access, typically through an API. Despite limited access, techniques like query-based optimization [7] allow adversaries to mount effective attacks.

2.2.2. Adversary's Goals

The objectives of adversarial attacks can be broadly classified into:

- **Confidentiality attacks** aim to extract sensitive information from the model or its training data.
- **Integrity attacks** seek targeted misclassification while leaving other predictions unaffected. For instance, an autonomous vehicle misclassifying stop signs as speed limit signs can have catastrophic consequences [4].
- **Availability attacks** degrade model performance globally, rendering the system unusable or unreliable.

2.3. Evaluation Metrics

Evaluation metrics in adversarial machine learning serve as critical tools to measure the effectiveness of attacks and the robustness of defense mechanisms. These metrics fall into three primary categories: (i) attack success, (ii) degradation of model performance, and (iii) computational efficiency. Table 1 summarizes the most commonly used metrics in each category, providing a basis for consistent benchmarking across studies.

These metrics guide both attack development and defense design. Targeted and untargeted success rates capture the precision and aggressiveness of adversarial examples. Accuracy drop and robustness bounds help assess generalizability and worst-case behavior, while query and time complexity are especially relevant in real-world deployments where computational cost and API restrictions are present. Certified defenses typically report formal robustness bounds, while empirical metrics such as CLEVER [18] and AutoAttack [21] provide rigorous and reproducible evaluations under standardized threat models.

2.4. Datasets and Benchmarks

Robust and reproducible evaluation in adversarial machine learning heavily relies on standardized datasets and benchmark frameworks. These datasets enable comparative analysis of attack and defense strategies under consistent settings.

Table 1: Evaluation Metrics in Adversarial Machine Learning

Category	Metric	Description and Use
Attack Effectiveness	Targeted Attack Success Rate (ASR) [16]	Percentage of adversarial inputs that cause the model to output a specific incorrect class.
	Untargeted ASR [2]	Measures the percentage of adversarial inputs that cause any incorrect classification.
Model Performance	Accuracy Drop [17]	Difference in clean test accuracy before and after attack exposure. Indicates global robustness loss.
	Robustness Bounds [18, 19]	Measures provable or empirical guarantees under bounded perturbations. Includes CLEVER scores and certified defenses.
Efficiency	Query Efficiency [20]	Number of queries to the target model needed to craft a successful adversarial example (relevant in black-box attacks).
	Time Complexity [21]	Computation time to craft adversarial samples (can include both offline and online phases).

Table 2: Representative Datasets and Benchmarks for Adversarial Machine Learning

Domain	Name	Description and Use
Image Classification	MNIST [22]	Handwritten digits (28x28). Used for evaluating simple perturbation-based attacks.
	CIFAR-10/100 [23]	Color images (32x32). Popular for evaluating transferability and defenses.
	ImageNet [24]	Large-scale dataset with 1000 classes. Used for high-complexity attacks and defenses.
NLP	SQuAD [25]	Question answering benchmark; adversarially modified queries test model understanding.
	GLUE [26]	NLP benchmark covering sentiment, entailment, and similarity.
Network Security	NSL-KDD [27]	Benchmark for intrusion detection; used for adversarial evasion in anomaly detection.
	UNSW-NB15 [28]	Modern attack traffic and normal activities for network intrusion research.
Malware Analysis	EMBER [29]	Portable executable (PE) files with labeled malware/benign examples.
	Microsoft Malware Challenge [30]	Dataset for static malware classification tasks.
Face Recognition	LFW [31]	Face verification benchmark with aligned image pairs.
	CelebA [32]	Large-scale dataset with facial attributes; used for poisoning and attribute manipulation.
Benchmarks	RobustBench [33]	Leaderboard for adversarial robustness across standardized threat models.
	AutoAttack [21]	Parameter-free ensemble of attacks for robust evaluation.
	CLEVER [18]	Metric-based framework for estimating robustness bounds.

Table 2 categorizes commonly used datasets and robustness benchmarks by domain, alongside their typical use cases and relevance in adversarial research.

These datasets and benchmarks collectively facilitate the development, evaluation, and comparison of robust machine learning models. Vision-based benchmarks such as ImageNet and CIFAR serve as primary testbeds for attack generalization and certified defenses. NLP datasets like SQuAD and GLUE are increasingly used to study textual perturbations and cross-modal vulnerabilities. Meanwhile, domain-specific datasets from security (e.g., EMBER, NSL-KDD) and privacy-sensitive applications help expose real-world adversarial concerns and drive practical defense innovation.

2.5. Adversarial Examples and Their Generations

Adversarial examples constitute a fundamental security threat to machine learning systems by enabling attackers to intentionally manipulate model predictions through carefully crafted input perturbations. Unlike random noise or benign input variations, adversarial perturbations are optimized to exploit vulnerabilities in a model's learned decision boundaries, often remaining imperceptible to human observers while inducing incorrect or highly confident mispredictions. As a result, adversarial examples undermine the reliability, safety, and trustworthiness of machine learning models deployed in security- and safety-critical applications, including autonomous driving, medical diagnosis, fraud detection, and intrusion detection systems.

From a threat-model perspective, adversarial examples exacerbate the inherent asymmetry between attackers and defenders. Attackers only need to identify a single effective perturbation to compromise model behavior, whereas defenders must anticipate and defend against a wide range of possible attack strategies, access assumptions, and perturbation constraints. Moreover, adversarial examples can be generated under varying levels of attacker knowledge, ranging from full access to model parameters and gradients (white-box setting) to highly restrictive scenarios where the attacker can only query model outputs (black-box setting). This flexibility significantly lowers the barrier to real-world exploitation.

At a high level, the generation of adversarial examples is commonly formulated as an optimization problem that seeks to maximize the model's prediction error while constraining the perturbation magnitude. Given an input–label pair (x, y), a trained model f_θ with parameters θ, and a loss function $\mathcal{L}(\cdot, \cdot)$,

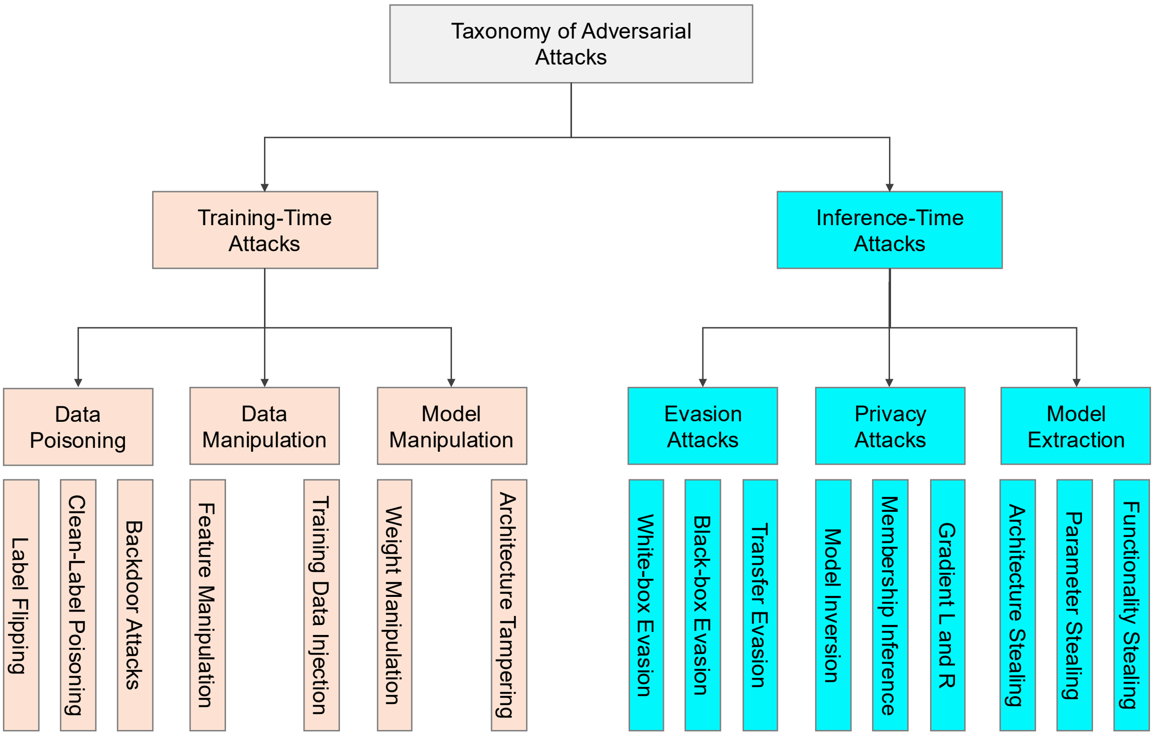

Figure 2: Taxonomy of adversarial attacks on machine learning systems.

adversarial example generation can be expressed as:

$$x' = \arg \max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}(f_\theta(x'), y), \tag{1}$$

where $\mathcal{B}_\epsilon(x)$ denotes a constraint set that limits the perturbation applied to the original input, typically defined using an ℓ_p norm. This formulation captures the core objective shared by many adversarial attack methods, regardless of their specific implementation.

Based on attacker capabilities and optimization strategies, adversarial example generation methods can be broadly categorized into several classes. In white-box settings, gradient-based attacks directly exploit the differentiability of deep neural networks to efficiently compute perturbations using loss gradients. Iterative variants further refine adversarial inputs through repeated optimization steps, yielding stronger and more transferable attacks. Optimization-based methods extend this approach by explicitly balancing misclassification objectives with perturbation minimization, often producing highly effective adversarial examples with small distortion.

In black-box scenarios, where model gradients and parameters are inaccessible, adversarial examples can still be generated through query-based or decision-based methods. These approaches rely on probing model outputs to estimate gradients, approximate decision boundaries, or iteratively adjust perturbations based on feedback signals. In addition, the transferability property of adversarial examples allows attackers to craft adversarial inputs using surrogate models and successfully apply them to unseen target models, highlighting the existence of shared vulnerabilities across different architectures and training procedures.

Overall, the diversity of adversarial example generation techniques demonstrates that adversarial threats are not confined to specific models, tasks, or access assumptions, but instead reflect a systemic vulnerability in modern machine learning systems. Understanding both the security implications of adversarial examples and the fundamental mechanisms underlying their generation is essential for evaluating model robustness and motivating the attack and defense strategies discussed in the remainder of this survey.

2.6. *Taxonomy of Adversarial Attacks*

Adversarial attacks on machine learning systems are commonly classified by the stage of the learning pipeline they target. These stages include the training phase and the inference phase. Such categorization is essential for understanding the attack surface and developing appropriate countermeasures. Training-time attacks aim to compromise the integrity of a model during its learning phase, embedding persistent vulnerabilities that can be exploited after deployment.

Figure 2 presents a structured taxonomy of adversarial attacks against machine learning models, categorized by the stage of the ML pipeline they target. Training-time attacks are grouped into data poisoning, data manipulation, and model manipulation, reflecting how adversaries can corrupt the learning process. Inference-time attacks include evasion attacks, privacy attacks (e.g., membership inference and model inversion), and model extraction threats such as functionality stealing. Each category is further subdivided into representative attack strategies to highlight the breadth of adversarial techniques that compromise model performance, confidentiality, and trustworthi-

Table 3: Taxonomy of Training-Time Attacks on Machine Learning Systems

Attack Category	Type	Description
Data Poisoning	Label Flipping [34, 35, 36]	Training labels are flipped without changing input features, shifting decision boundaries and degrading model performance.
	Clean-Label Poisoning [37]	Crafted benign-looking inputs with correct labels mislead the model into incorrect inference behavior.
	Backdoor Attacks [38, 39]	Trigger patterns are embedded in training data to elicit malicious behavior under specific conditions.
Data Manipulation	Feature Manipulation [40, 41, 35]	Specific input features are altered to bias the learning process while preserving labels and realism.
	Training Data Injection [42]	Adversarial samples with natural appearance are injected to shape model behavior undetectably.
Model Manipulation	Weight Manipulation [43, 44]	Model weights are directly altered to embed backdoors or logic bombs that persist through fine-tuning.
	Architecture Tampering [45, 46, 47]	Structural components such as layers or skip connections are modified to introduce hidden vulnerabilities.

ness. This taxonomy provides a conceptual framework for analyzing threat vectors and guiding the development of corresponding defense strategies.

3. Training-Time Attacks

As shown in Figure 3, the figure provides an intuitive illustration of how adversarial attacks can be mounted at different stages of the machine learning lifecycle, encompassing both training-time and inference-time threat models. In this section, we first discuss training-time attacks, followed by a detailed examination of inference-time attacks in the next section.

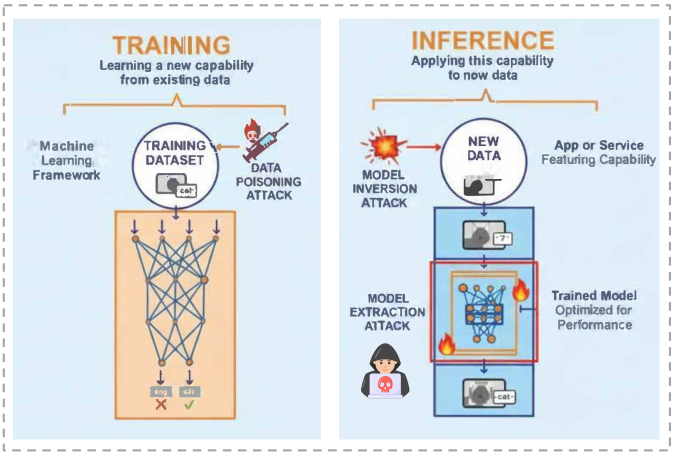
TRAINING
Learning a new capability from existing data
Machine Learning Framework
TRAINING DATASET
cat
DATA POISONING ATTACK
INFERENCE
Applying this capability to now data
NEW DATA
App or Service Featuring Capability
MODEL INVERSION ATTACK
7
MODEL EXTRACTION ATTACK
Trained Model Optimized for Performance
cat

Figure 3: Examples of adversarial attacks targeting different stages of the machine learning pipeline, including training-time data poisoning and inference-time model inversion and model extraction attacks.

Training-time attacks pose a particularly insidious threat to machine learning systems. By tampering with the training data, altering the learning process, or modifying the model architecture, adversaries can embed malicious behaviors that are difficult to detect through standard validation techniques. These attacks exploit the inherent learning dynamics of the model by shifting decision boundaries, corrupting internal representations, or manipulating optimization pathways. Once deployed, the compromised model typically performs as expected on benign inputs but produces erroneous or malicious outputs when exposed to carefully crafted trigger conditions introduced by the attacker. As shown in Table 3, we detail several types of training-time attacks, categorized broadly into data poisoning, data manipulation, and model manipulation.

3.1. *Data Poisoning Attacks*

Data poisoning attacks aim to corrupt the training process by injecting malicious examples into the dataset. These attacks can degrade model performance globally or embed targeted backdoors that trigger incorrect behavior under specific conditions. Poisoning techniques include label flipping, feature manipulation, and clean-label poisoning, each exploiting different aspects of the learning algorithm. Such attacks are particularly dangerous in open or crowdsourced training pipelines, where data curation is minimal.

3.1.1. *Label Flipping*

Label flipping is among the most basic and effective poisoning techniques. The adversary alters the labels of existing training samples without modifying their features. These manipulated samples are indistinguishable from legitimate data to human reviewers or standard validation procedures. This subtlety makes detection challenging.

Label flipping attacks distort the model's decision boundaries. In binary classification, flipping labels near the boundary can significantly shift the learned parameters. Xiao et al. [34] demonstrated that altering just 20% of labels in sensitive regions can lead to a 35% reduction in classification accuracy on clean test data. More sophisticated variants [35, 36] apply influence function analysis and gradient-based optimization to locate high-impact points for label alteration. Auxiliary models also help identify optimal flipping targets, enhancing both stealth and efficacy.

Label flipping can be modeled as constructing a poisoned dataset $\mathcal{D}'$ by changing the labels of a selected subset of training points while keeping their inputs unchanged. Let $\mathcal{D}$ =

$\{(x_i, y_i)\}_{i=1}^N$ and let $\mathcal{S} \subseteq \{1, \ldots, N\}$ denote the indices chosen for poisoning. The poisoned dataset is

$$\mathcal{D}' = \{(x_i, \tilde{y}_i)\}_{i=1}^N, \quad \tilde{y}_i = \begin{cases} \pi(y_i), & i \in \mathcal{S}, \\ y_i, & i \notin \mathcal{S}, \end{cases} \tag{2}$$

where $\pi(\cdot)$ denotes a label-mapping rule (e.g., random flipping or targeted flipping). Training on $\mathcal{D}'$ yields parameters $\theta' = \arg\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}'}[\mathcal{L}(f_\theta(x), y)]$, illustrating how label corruption directly perturbs the empirical risk being minimized and can shift the learned decision boundary.

3.1.2. Clean-Label Poisoning

Clean-label poisoning attacks insert adversarial samples into the training dataset with correct labels, making them especially deceptive. These inputs are crafted to appear visually or semantically benign, thus evading detection, but are optimized to induce incorrect predictions when specific triggers or features are encountered at inference time.

This approach is particularly dangerous in transfer learning scenarios, where pretrained models are fine-tuned on small, potentially vulnerable datasets. For example, in computer vision, adversaries might subtly modify images of dogs to resemble cats while keeping the label "dog," thereby manipulating feature extraction layers. Shafahi et al. [37] demonstrated that poisoning just 50 out of 50,000 training samples in CIFAR-10 could induce targeted misclassification in over 60% of test cases, without affecting clean test accuracy, highlighting the stealth and potency of this attack vector.

Clean-label poisoning typically aims to induce a targeted error on a specific test-time target input x_t, while keeping injected samples correctly labeled. A common abstraction is to optimize poison samples to maximize the loss on (x_t, y_t) after training:

$$\max_{\mathcal{D}_{\text{adv}}} \mathcal{L}(f_{\theta'(\mathcal{D}\cup\mathcal{D}_{\text{adv}})}(x_t), y_t) \quad \text{s.t.} \quad (x, y) \in \mathcal{D}_{\text{adv}} \Rightarrow y = y_{\text{c}}, \tag{3}$$

where $\theta'(\cdot)$ denotes the parameters obtained by training on the poisoned dataset. This formulation highlights the core challenge of clean-label poisoning: the attacker must shape the learned representation using seemingly benign, correctly labeled points, so that the target input is misclassified at inference time without noticeably harming standard test accuracy.

3.1.3. Backdoor Attacks

Backdoor attacks embed hidden patterns (triggers) into the model during training, enabling adversaries to induce malicious behavior at inference time when the trigger is present. In the absence of the trigger, the model behaves normally, which makes these attacks highly stealthy and difficult to detect using standard evaluation methods. This duality between clean and triggered behavior poses a serious threat in real-world deployments.

The attack typically involves two phases: a poisoning phase, where trigger-laden inputs are inserted into the training data with attacker-specified labels, and an activation phase, where the model is queried with inputs containing the trigger. Triggers can take various forms, including visual artifacts (e.g., small stickers on traffic signs), specific keywords in natural language text, or imperceptible frequency patterns in audio. Gu et al. [38] demonstrated that simple pixel-level triggers can create highly effective backdoors in image classifiers. Liu et al. [39] further showed that even internal neuron activations can be manipulated to embed Trojan behavior. Notably, backdoors can survive common model modifications such as pruning, quantization, and fine-tuning, highlighting the need for specialized detection and mitigation strategies.

A backdoor attack introduces a trigger transformation $\mathcal{T}(\cdot)$ (e.g., a small patch, keyword, or pattern) such that inputs containing the trigger are mapped to an attacker-chosen target label y_b. Training can be viewed as minimizing a mixture of clean and backdoor objectives:

$$\min_\theta (1-\alpha)\,\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f_\theta(x), y)] + \alpha\,\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f_\theta(\mathcal{T}(x)), y_b)], \tag{4}$$

where α is the poisoning ratio. After training, the attacker expects $f_\theta(x) \approx y$ for benign inputs, but $f_\theta(\mathcal{T}(x)) \approx y_b$ when the trigger is present, capturing the dual behavior that makes backdoors difficult to detect through standard evaluation.

3.2. Data Manipulation

Data manipulation attacks compromise the integrity of the training dataset by injecting or modifying samples to influence the model's behavior. Unlike label flipping, these attacks alter the input features while preserving labels to remain undetected. Feature perturbations are often subtle and optimized to exploit model inductive biases, leading to systematic errors under targeted conditions. These attacks are particularly effective in settings where data collection is distributed or weakly supervised.

3.2.1. Feature Manipulation

Feature manipulation attacks aim to alter specific attributes of the training data while preserving the associated labels. Unlike label flipping or backdoor attacks, the adversary introduces subtle changes to input features, which can distort the model's learning process and induce misclassifications in targeted regions of the input space. These attacks are particularly potent when the adversary possesses partial knowledge of the model's architecture, preprocessing pipeline, or feature extraction mechanisms.

Zhang et al. [40] demonstrated that strategically manipulated features could cause trained models to systematically fail under specific conditions, even when standard training metrics indicate normal behavior. More recent techniques such as Witches' Brew [41] and MetaPoison [35] apply gradient-based optimization to craft poison instances that remain visually or semantically indistinguishable from clean data. These modern approaches enhance stealth by ensuring that the manipulated features lie within the natural data manifold and are optimized to evade detection during manual inspection or automated validation. Feature manipulation poses a unique threat in domains like computer vision and natural language processing, where high-dimensional data and complex preprocessing pipelines offer rich opportunities for subtle but effective attack vectors.

Feature manipulation can be modeled as crafting small perturbations δ_i to selected training inputs while preserving labels:

$$x_i' = x_i + \delta_i, \quad \|\delta_i\|_p \leq \epsilon, \quad y_i' = y_i, \tag{5}$$

where the perturbations are optimized so that training on $\{(x_i', y_i)\}$ biases the learned model toward attacker-desired errors, while keeping poisoned samples visually or semantically plausible.

3.2.2. Training Data Injection

Training data injection refers to the process of introducing maliciously crafted examples into the training dataset to subtly manipulate the model's behavior. Unlike more overt poisoning techniques, such as label flipping or disruptive feature corruption, this approach maintains the statistical coherence and semantic plausibility of the training data, making detection significantly more challenging.

Recent developments in generative modeling have enabled attackers to create high-fidelity, task-aligned examples that can be injected into the training corpus. These samples are carefully constructed to influence model predictions in targeted ways without degrading overall performance. For example, in natural language processing (NLP), attackers can introduce syntactically and semantically valid sentences that encode hidden biases or trigger behaviors. Turner et al. [42] demonstrated that injecting as little as 0.1% of such examples into the training set can lead to the emergence of persistent and exploitable model vulnerabilities, all while maintaining high test accuracy on standard benchmark datasets. Training data injection thus poses a potent threat, especially in large-scale data collection scenarios where manual curation is impractical.

3.3. Model Manipulation

Model manipulation attacks target the internal components or training procedures of machine learning models to implant hidden behaviors or degrade performance. Unlike data-centric poisoning, these attacks may alter initialization schemes, introduce malicious layers, or tamper with optimization routines. Their stealthy nature allows compromised models to behave normally during evaluation while exhibiting adversarial behaviors under specific conditions.

3.3.1. Weight Manipulation

Weight manipulation attacks involve the direct alteration of a model's internal parameters, which are typically the learned weights of neural networks, with the intent of embedding malicious behavior. These attacks are particularly salient in distributed and federated learning environments, where the training process is decentralized and thus more susceptible to tampering [43].

Such manipulations can be subtle, targeting only a small subset of model weights while preserving performance on standard test cases. Nonetheless, these minimal changes can implant backdoors, biases, or decision logic that persist through downstream fine-tuning. This resilience makes them difficult to detect using conventional validation methods. In some cases, adversaries use optimization-based approaches to find minimal weight adjustments that produce targeted misclassifications [44].

Weight manipulation can be abstracted as finding a small parameter perturbation $\Delta\theta$ that preserves clean accuracy while enforcing a malicious behavior:

$$\min_{\Delta\theta} \|\Delta\theta\| \quad \text{s.t.} \quad f_{\theta+\Delta\theta}(x) \approx f_\theta(x) \text{ for } x, \quad f_{\theta+\Delta\theta}(x_b) = y_b, \tag{6}$$

where (x_b, y_b) represents a trigger condition or targeted behavior imposed by the attacker.

3.3.2. Architecture Tampering

Architecture tampering attacks involve the deliberate manipulation of a machine learning model's structural design to embed covert vulnerabilities. Instead of modifying the training data or parameters, these attacks alter the model's topology such as by inserting hidden pathways, modifying activation functions, or restructuring residual connections. The changes are engineered to preserve performance on standard evaluation metrics while enabling malicious behavior under specific conditions.

Recent research has demonstrated the feasibility of such attacks. For instance, adversaries may implant logic bombs by appending inactive sub-networks that activate only when specific trigger patterns are present [45]. Other techniques modify internal skip connections or reuse redundant branches to achieve bifurcated behavior [46]. These manipulations are especially insidious in settings involving neural architecture search (NAS), where attackers can influence automated architecture generation pipelines to inject tampered designs [47].

Because architectural modifications often evade conventional auditing methods focused on weights or training data, architecture tampering represents an emergent and critical threat vector. Addressing this challenge will require model certification techniques that analyze both the structure and functionality of neural networks.

4. Inference-Time Attacks

Inference-time attacks target deployed machine learning models during prediction rather than training. Unlike training-phase attacks, these threats do not require access to the training data or process, making them significantly more practical in real-world scenarios. As ML models are widely deployed in critical applications like healthcare, finance, and autonomous systems, these attacks pose serious privacy, safety, and reliability concerns. Table 4 provides a taxonomy of the primary types of inference-time attacks, which we detail in this section.

4.1. Evasion Attacks

Evasion attacks occur during the inference phase, where adversaries craft carefully perturbed inputs to mislead the model without altering its internal parameters. These attacks exploit the model's sensitivity to small, often imperceptible, changes in input data, resulting in incorrect or manipulated predictions.

Table 4: Taxonomy of Inference-Time Attacks on ML Systems

Category	Subtype	Description
Evasion Attacks	White-box [2, 17, 16]	Use full model knowledge to compute adversarial perturbations that mislead predictions.
	Black-box [48, 49, 20]	Query-only attacks that estimate gradients or exploit decision boundaries to craft inputs.
	Transfer-based [50, 51, 52]	Leverage transferability of adversarial examples across different models.
Privacy Attacks	Model Inversion [53, 54]	Reconstruct training samples using output probabilities and gradients.
	Membership Inference [55, 56]	Infer if a particular input was part of a model's training set.
Model Extraction	Architecture Stealing [57, 58]	Recover the target model's architectural design using timing and output analysis.
	Parameter Stealing [59, 60, 61]	Reconstruct model weights using gradient estimation or equation solving.
	Functionality Stealing [62, 63]	Imitate a model's decision boundary via knowledge distillation or surrogate training.

Evasion techniques are categorized based on the adversary's knowledge: white-box attacks assume full access to the model's architecture and gradients, black-box attacks operate solely via input-output queries, and transfer attacks leverage adversarial examples generated on substitute models. Understanding these categories is critical for designing robust and generalizable defenses.

4.1.1. White-box Attacks

White-box attacks assume complete access to the model architecture, parameters, and gradients. In this setting, adversaries can directly optimize perturbations with respect to the model's loss function to generate inputs that induce misclassification.

The *Fast Gradient Sign Method* (FGSM) [2] is one of the earliest and most influential gradient-based attacks. It perturbs an input x along the sign of the gradient of the loss function $J(\theta, x, y)$ with respect to the input:

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

where ϵ controls the perturbation magnitude. While computationally efficient, FGSM performs only a single gradient step, which may lead to suboptimal perturbations and limited attack success under tighter perturbation constraints.

To overcome this limitation, the *Basic Iterative Method* (BIM) [64] and its generalization, *Projected Gradient Descent* (PGD) [65], apply FGSM iteratively. BIM accumulates multiple small perturbations, refining adversarial strength while maintaining imperceptibility. PGD extends this approach by introducing projection back onto the ℓ_∞-ball after each iteration:

$$x_{adv}^{t+1} = \Pi_{x+\mathcal{S}} \left(x_{adv}^{t} + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_{adv}^{t}, y)) \right)$$

PGD is widely regarded as a universal first-order adversary and serves as a benchmark for evaluating model robustness in many defense studies.

Beyond gradient-sign attacks, optimization-based methods such as the *Carlini & Wagner (C&W)* attack [16] reformulate the adversarial generation process as a constrained optimization problem that minimizes perturbation distortion while enforcing misclassification. This method achieves high attack success and produces perturbations that are often imperceptible to humans.

Several other seminal attacks contribute to the evolution of white-box evasion research. The *DeepFool* algorithm [66] computes minimal perturbations by iteratively linearizing the classifier's decision boundary, yielding efficient and nearly imperceptible adversarial examples. The *Jacobian-based Saliency Map Attack (JSMA)* [67] instead leverages the model's Jacobian to identify input features most influential to the output class, selectively modifying a few key pixels to induce misclassification.

Overall, these white-box attacks form the foundation of adversarial research, illustrating both the diversity of optimization objectives (e.g., norm minimization, saliency targeting, and minimal perturbation) and the evolution from single-step to iterative and optimization-based strategies. They continue to serve as standard benchmarks for evaluating the robustness of modern deep learning models.

4.1.2. Black-box Attacks

In black-box settings the attacker lacks access to model internals (architecture, parameters, or gradients) and must rely solely on input–output queries to craft adversarial examples. Black-box attacks are typically classified by the type of feedback available (score-based, decision-based) and by their query strategy (gradient approximation, surrogate training, or decision-boundary search).

Gradient-free gradient estimation. Zeroth-Order Optimization (ZOO) [48] approximates gradients via finite differences:

$$\frac{\partial f}{\partial x_i} \approx \frac{f(x + he_i) - f(x)}{h},$$

and uses these estimates to perform iterative optimization. While effective, ZOO is computationally expensive and query-intensive in high-dimensional spaces. Subsequent work improves query efficiency by estimating gradients in low-dimensional subspaces or using population-based estimators such as Natural Evolution Strategies (NES) [20], which sample perturbations from a distribution and average directional derivatives to reduce the number of queries.

Decision-based and score-based attacks. Decision-based attacks (e.g., the Boundary Attack [49]) operate when the model only returns class labels. The Boundary Attack begins from a known adversarial example and performs a random walk toward the target input while maintaining the adversarial property, making it effective in the label-only setting but often requiring many queries. Score-based attacks exploit access to confidence or probability scores to guide perturbations more efficiently than purely decision-based methods. Meanwhile, the *One-Pixel Attack* [68] demonstrates that even a single-pixel change, optimized via differential evolution, can fool deep neural networks, highlighting the extreme sensitivity of model decision boundaries.

Surrogate (substitute) model and transfer approaches. An alternative black-box strategy trains a surrogate (or *substitute*) model to mimic the target's input–output behavior and then crafts white-box adversarial examples on the surrogate that transfer to the target [7]. Transfer-based attacks are particularly practical when query budgets are limited, although their success depends on transferability between models and on how well the surrogate approximates the target's decision boundaries.

Query-efficiency and optimization heuristics. Practical black-box attacks emphasize query efficiency. Techniques include (i) estimating gradients in lower-dimensional bases, (ii) using bandit or population-based optimizers (NES), and (iii) adopting simultaneous perturbation methods (which perturb multiple coordinates at once) to reduce per-iteration queries. Attackers often combine these tactics with input priors (e.g., image structure) to further reduce query counts.

Practical considerations and defenses. Black-box attacks are typically more query- and time-intensive than white-box attacks, and their real-world feasibility depends on API rate limits, detection mechanisms, and query costs. Defenses such as output-limited APIs (returning only top-1 labels), rate limiting, response randomization, and anomaly-based query detectors are effective mitigations but must be balanced against utility and usability requirements.

Overall, black-box methods broaden the threat model by showing that even limited-feedback deployments are vulnerable. They complement transfer and white-box attacks in robustness evaluations and motivate defenses that consider query-based adversaries and practical deployment constraints.

4.1.3. Transfer Attacks

Transfer-based attacks exploit the empirical phenomenon that adversarial examples crafted for one model (a surrogate or substitute) often remain effective against other models, even those with different architectures, random initializations, or training datasets. Transferability enables practical black-box attacks when direct access to the target model is unavailable, and therefore it represents a serious threat to deployed systems.

A common approach to improve transferability is to stabilize and diversify the optimization process used to craft perturbations. The *Momentum Iterative Method* (MIM) [50] augments iterative gradient updates with a momentum term to accumulate gradient directions across steps:

$$g_{t+1} = \mu g_t + \frac{\nabla_x J(\theta, x_t, y)}{\|\nabla_x J(\theta, x_t, y)\|_1}, \qquad x_{t+1} = x_t + \alpha \cdot \mathrm{sign}(g_{t+1}),$$

where μ is the momentum factor and α the step size. By smoothing the update trajectory, MIM reduces oscillation and produces perturbations that generalize better across models, making it particularly effective in transfer-based black-box scenarios.

Another influential class is *Universal Adversarial Perturbations* (UAPs) [51, 52], which seek a single input-agnostic perturbation δ that causes widespread misclassification:

$$\text{Find } \delta \text{ such that } f(x+\delta) \neq f(x),\ \forall x \in \mathcal{X},\ \text{and } \|\delta\|_p \leq \xi.$$

UAPs reveal that certain directions in high-dimensional input space are broadly effective at altering model outputs, underscoring shared vulnerabilities across independently trained networks.

Several factors influence transferability, including model architecture similarity, shared training data or preprocessing, input representation, and the attack objective (targeted vs. untargeted). Empirically, perturbations crafted to target low-frequency or semantically relevant features often transfer more reliably than those that exploit high-frequency noise. Techniques such as input transformation during attack generation (e.g., random resizing, translation) and ensemble-based surrogate training also improve cross-model success by producing perturbations that are robust to variation.

Defenses against transfer attacks typically aim to reduce shared weaknesses across models. In robustness evaluations, transfer attacks are a crucial component of a comprehensive threat model because they capture realistic adversarial scenarios where the attacker has only oracle access or can train a surrogate. When reporting transfer-based results, it is important to (i) specify surrogate model families and training data, (ii) evaluate both targeted and untargeted transfer success rates, and (iii) test across multiple target architectures to avoid overestimating robustness against a narrow set of attacks. Overall, transfer-based attacks highlight systemic, cross-model vulnerabilities and motivate defenses that prioritize representational diversity, robust training, and careful evaluation protocols to ensure resilience under realistic black-box threat models.

4.2. Privacy Attacks

Privacy attacks target the confidentiality of machine learning systems by attempting to extract sensitive information about training data, model parameters, or user inputs. These attacks expose hidden privacy risks in AI services and motivate the development of privacy-preserving mechanisms. The most studied types include *model inversion*, *membership inference*, and *gradient leakage*. Each highlights distinct vulnerabilities depending on what information the attacker can access and how the model exposes internal or output data.

4.2.1. Model Inversion

Model inversion attacks aim to reconstruct or approximate private training inputs based on the information exposed through model outputs. Fredrikson et al. [53] demonstrated that confidence scores from a facial recognition classifier could be used to iteratively recover approximate facial features of individuals in the training set. By optimizing input candidates to maximize a target class probability, attackers can gradually generate synthetic inputs visually similar to genuine samples. This poses severe risks in sensitive applications such as biometric authentication or healthcare diagnostics, where reconstructed images or attributes could reveal personally identifiable information.

More advanced approaches leverage generative modeling to improve reconstruction fidelity. Yang et al. [54] utilized Generative Adversarial Networks (GANs) to learn an inverse mapping from model outputs (e.g., logits or gradient information) back to plausible input representations. These methods generalize beyond simple classifiers and can recover semantically coherent data even from limited side information. Hybrid attacks combining GAN priors with gradient-based refinement have achieved high-quality reconstructions of images and tabular records, underscoring the growing threat of inversion-based leakage in modern ML deployments.

Model inversion can be formalized as reconstructing an input x that maximizes the model's confidence for a target class y_t (or matches a desired output statistic). In its simplest score based form, the attacker solves

$$x^\star = \arg\max_{x \in \mathcal{X}} \; p_\theta(y_t \mid x) \; - \; \lambda R(x), \tag{7}$$

where $p_\theta(y_t \mid x)$ denotes the target class probability returned by the model, $R(x)$ is a regularizer that encodes a prior over plausible inputs (for example, an ℓ_2 penalty, total variation, or a generative prior), and λ balances fidelity and realism.

When auxiliary knowledge is available (for example, partial attributes or a public prior), inversion can be posed as matching model outputs to a desired vector s (such as logits or confidence scores):

$$x^\star = \arg\min_{x \in \mathcal{X}} \; \|g_\theta(x) - s\|_2^2 \; + \; \lambda R(x), \tag{8}$$

where $g_\theta(x)$ represents the exposed output (logits, probabilities, or intermediate signals). These objectives capture why confidence scores and rich output APIs significantly increase inversion risk.

4.2.2. Membership Inference

Membership inference attacks determine whether a specific data record was included in a model's training dataset. This form of attack threatens data confidentiality in scenarios such as medical or financial prediction services. The core idea is that models often exhibit higher confidence or lower loss values on training data than on unseen samples.

Shokri et al. [55] introduced a seminal shadow-model framework, where attackers train multiple local models on synthetic datasets to mimic the target's decision behavior. By comparing output confidence distributions for known members and non-members, they train a meta-classifier to infer membership. This approach was later simplified by Salem et al. [56], who showed that shadow models are not essential: simple thresholding on confidence or entropy values can still reveal membership with high accuracy. These results demonstrate that even limited black-box access can leak information about individual training records.

Subsequent research has extended membership inference to a variety of settings, including federated learning, differential privacy, and large-scale foundation models. Attackers can exploit gradients shared during distributed training or confidence-based overfitting in fine-tuned models. Moreover, correlation-based membership inference can expose user participation in multi-party datasets, raising critical privacy concerns for collaborative learning systems.

Membership inference can be modeled as a binary hypothesis test for a record (x, y):

$$H_0 : (x, y) \notin \mathcal{D} \quad \text{vs.} \quad H_1 : (x, y) \in \mathcal{D}.$$

A common and effective black box strategy uses the observation that training points typically have smaller loss. The attacker predicts membership using a threshold rule

$$\hat{m}(x, y) = \mathbb{I}\left[\mathcal{L}(f_\theta(x), y) \le \tau\right], \tag{9}$$

where τ is chosen using auxiliary data (or calibrated via shadow models), and $\mathbb{I}[\cdot]$ is the indicator function.

Equivalently, attacks may threshold confidence, entropy, or margin. For example, using predictive entropy $H(p_\theta(\cdot \mid x))$:

$$\hat{m}(x) = \mathbb{I}\left[H(p_\theta(\cdot \mid x)) \le \tau_H\right]. \tag{10}$$

These formulations make explicit that membership leakage is driven by overfitting and calibration gaps, since the decision rule exploits systematic differences between member and non member outputs.

4.2.3. Gradient Leakage and Reconstruction

A more recent class of privacy attacks, gradient leakage, directly exploits gradients exchanged during distributed or federated learning to recover sensitive training data. Zhu et al. [69] demonstrated that even a single gradient vector could enable the near-exact reconstruction of private input samples through iterative optimization. This vulnerability arises because gradients implicitly encode information about both feature values and labels.

Follow-up work has enhanced reconstruction fidelity through regularization and prior knowledge about data distribution, leading to practical gradient inversion frameworks such as DLG and iDLG. These attacks highlight that sharing gradients or parameter updates, even without direct data exchange, may still compromise user privacy. Accordingly, combining secure aggregation protocols with differential privacy noise has become a critical direction for protecting distributed learning systems.

In gradient leakage attacks, the adversary observes a gradient (or update) g produced during training, for example

$$g = \nabla_\theta \mathcal{L}(f_\theta(x), y),$$

and attempts to recover the private training example (x, y) that generated it. A widely used abstraction reconstructs inputs by solving a gradient matching problem:

$$(x^\star, y^\star) = \arg \min_{x \in \mathcal{X},\, y \in \mathcal{Y}} \|\nabla_\theta \mathcal{L}(f_\theta(x), y) - g\|_2^2 \; + \; \lambda R(x), \tag{11}$$

where $R(x)$ is an input prior (for example, total variation for images or language model priors for text), and λ controls the strength of the prior.

In federated or distributed learning, attackers may observe an aggregated update $\Delta\theta$ or mini batch gradient $g_B = \sum_{i\in B} \nabla_\theta \mathcal{L}(f_\theta(x_i), y_i)$. Reconstruction then becomes a constrained optimization over a batch of inputs, which is why regularization, initialization strategies, and distribution priors can significantly improve attack fidelity in practice.

4.3. Model Extraction

Model extraction attacks aim to replicate a target model's behavior, parameters, or architecture through systematic querying or side-channel observation. By exploiting access to model predictions, attackers can reconstruct near-equivalent replicas without direct access to training data or source code. Such attacks threaten the intellectual property (IP) of commercial AI services, undermine model confidentiality, and can facilitate downstream attacks such as adversarial transfer or membership inference. Common approaches include *architecture stealing*, *parameter stealing*, and *functionality stealing*.

4.3.1. Architecture Stealing

Architecture stealing attacks seek to infer a model's underlying structure, such as depth, layer type, and activation functions, using only external observations. Oh et al. [57] demonstrated that side-channel information like response latency and output sensitivity can be exploited to approximate architectural details. By probing models with controlled inputs and analyzing timing variations or perturbation responses, attackers can reconstruct coarse-grained structural patterns. This is particularly concerning for cloud-hosted AI APIs, where the model's metadata is concealed but inference latency remains observable.

Jagielski et al. [58] extended this concept by employing neural architecture search (NAS) to automate model structure recovery through iterative black-box interaction. Their method progressively refines a candidate architecture based on similarity between predicted outputs, converging toward a replica with comparable functionality and topology. Such attacks demonstrate that even without source access, adversaries can recover significant architectural information, highlighting the need for architectural obfuscation and query randomization in commercial ML deployments.

4.3.2. Parameter Stealing

Parameter stealing focuses on recovering or approximating the internal weights of a trained model. Tramer et al. [59] first showed that models trained on public datasets can be cloned through repeated black-box queries and regression fitting, especially for shallow networks and linear classifiers. Their findings revealed that accurate approximations can be achieved with far fewer queries than expected, raising concerns about the confidentiality of deployed prediction APIs.

Building on this, Chandrasekaran et al. [61] demonstrated that incorporating prior knowledge of training dynamics, such as optimizer behavior, initialization schemes, or learning rate schedules, further improves reconstruction accuracy. In distributed and federated settings, Zhu et al. [60] showed that intercepting gradient updates during training allows attackers to reconstruct both parameters and data representations via inversion-based optimization. These results emphasize the importance of secure aggregation, model encryption, and gradient perturbation to safeguard parameter confidentiality.

4.3.3. Functionality Stealing

Functionality stealing replicates the predictive behavior of a target model by training a surrogate or student model based on query responses. Orekondy et al. [62] introduced the *Knockoff Nets* framework, which employs active learning to select informative queries that efficiently explore a target's decision boundaries. By leveraging feedback from the target's predictions, the attacker can train a student model achieving comparable accuracy with a limited number of adaptive queries.

Krishna et al. [63] advanced this idea through meta-learning, proposing a few-shot functionality stealing framework that learns generalizable imitation strategies. This allows adversaries to clone proprietary models even with minimal labeled data or limited query budgets. Such attacks have profound implications for commercial machine-learning-as-a-service (MLaaS) platforms, where prediction APIs expose valuable intellectual property. Effective defenses include query-rate limiting, model watermarking, output randomization, and access control mechanisms to detect and deter large-scale model replication.

5. Defense Mechanisms

Machine learning models deployed in critical applications are increasingly targeted by adversarial attacks. To secure these systems, a wide range of defense strategies have been developed, each aiming to protect different stages of the machine learning pipeline. These defenses are typically categorized into four primary layers: *preventive*, *detection*, *reactive*, and *privacy-preserving* mechanisms. Figure 4 provides a high-level schematic of defense mechanisms in adversarial machine learning, highlighting how different defense categories complement each other within a unified protection pipeline.

Table 5 provides a concise summary of these categories, including representative techniques and their core objectives. Each layer complements the others to form a holistic, multi-layered defense framework capable of countering diverse adversarial threats.

5.1. Preventive Defenses

Preventive defenses aim to strengthen machine learning models before adversarial inputs are encountered. Unlike detection or reactive strategies that respond after an attack, preventive approaches attempt to embed robustness into the model design or input preprocessing pipeline. They are typically applied during training or as front end transformations, proactively reducing a model's vulnerability. However, preventive defenses are not foolproof; attackers often adapt quickly to their structures and

Table 5: Taxonomy of Defense Mechanisms Against Adversarial Attacks

Defense Category	**Technique**	**Description**
Preventive Defenses	Input Sanitization [70, 71]	Removes adversarial perturbations through preprocessing techniques such as JPEG compression, smoothing, or projection onto clean data manifolds.
	Robust Training [17, 72]	Trains models with adversarial examples or optimization-based formulations to enhance resilience to perturbations.
	Certified Defenses [73, 74]	Provide provable robustness guarantees using techniques like randomized smoothing or convex relaxations.
Detection Mechanisms	Statistical Detection [75]	Identifies adversarial inputs using statistical deviations in input distributions or model confidence.
	NN-based Detection [76]	Trains auxiliary models to distinguish adversarial inputs from legitimate ones.
	Behavioral Analysis [77]	Monitors internal representations or temporal dynamics to flag anomalous behavior.
Reactive Defenses	Model Patching [78, 79]	Fine-tunes or reprograms specific model parameters to neutralize discovered vulnerabilities.
	Input Reconstruction [71, 80]	Uses autoencoders or projection methods to recover clean inputs.
	Ensemble Methods [81, 76]	Combines multiple models or detection strategies to improve robustness and mitigate attack transferability.
Privacy Preserving	Differential Privacy [82, 13]	Adds noise during training or inference to obscure individual data contributions.
	Federated Learning [83, 84]	Enables collaborative training without centralized access to sensitive data.
	Secure Aggregation [85, 83]	Aggregates model updates using cryptographic protocols to preserve individual privacy.

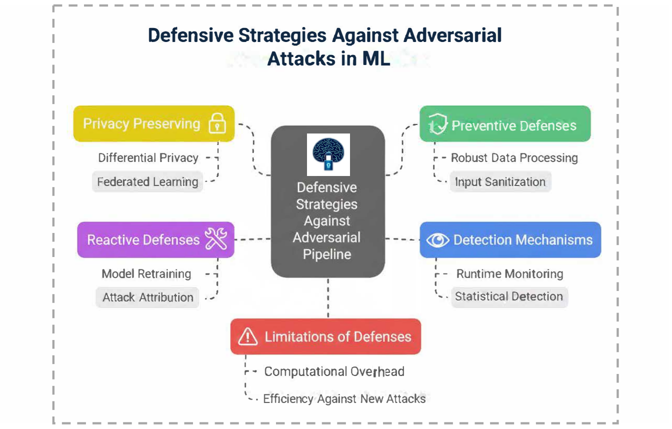

Figure 4: Overview of defensive strategies against adversarial attacks in machine learning, including preventive, detection, reactive, and privacy-preserving mechanisms.

assumptions, creating an ongoing cycle of refinement and circumvention. This section discusses three primary approaches, input sanitization, robust training, and certified defenses, and examines their practical benefits, inherent tradeoffs, and long term sustainability in real world settings.

5.1.1. *Input Sanitization*

Input sanitization removes or neutralizes adversarial perturbations before data are fed into the model. These methods assume that adversarial noise typically resides in imperceptible, high frequency components that can be filtered out while retaining the semantic content of the input. Classical sanitization techniques include JPEG compression, bit depth reduction, image quilting, and Gaussian blurring [70]. Such transformations are lightweight and model agnostic, making them attractive as plug in preprocessing defenses for computer vision and multimedia pipelines.

More advanced methods use learned projections to restore perturbed samples. Defense GAN [71], for instance, trains a generative model to project inputs onto the manifold of natural images by optimizing for a latent representation that reconstructs a clean version of the input. Autoencoder based approaches similarly use reconstruction loss to remove noise while preserving structural integrity. These techniques, however, introduce their own risks. Over reliance on learned sanitization can lead to reconstruction artifacts or domain overfitting, and attackers can craft adaptive perturbations that survive these transformations. Consequently, modern research emphasizes hybrid sanitization pipelines that combine hand crafted and learned methods, coupled with adaptive noise modeling and uncertainty estimation to sustain robustness against evolving attack strategies.

5.1.2. *Robust Training*

Robust training integrates adversarial awareness directly into the model optimization process and represents one of the most established preventive defenses in adversarial machine learning. The foundational adversarial training framework [2] augments clean data with adversarial examples generated during training, compelling the model to learn perturbation-invariant representations. While conceptually simple and empirically effective, this approach often incurs heavy computational costs and introduces a tradeoff between robustness and clean-data accuracy.

Building on this foundation, more advanced methods such as Projected Gradient Descent (PGD) adversarial training [17] iteratively generate stronger perturbations within bounded norms, while TRADES [72] formulates a theoretical compromise between natural accuracy and adversarial robustness via a dual-objective loss. Subsequent research, including interval bound propagation, curriculum-based adversarial training, and adversarial data augmentation, further enhances training efficiency and generalization.

Recently, a new wave of studies has emerged to advance robust training beyond conventional paradigms. These include adversarial training for single-modal architectures [86, 87, 88], few-shot adversarial training that leverages meta-learning for data-scarce scenarios [89, 90, 91, 92], adversarially robust knowledge distillation that transfers robustness from teacher to student networks [89, 93, 92], and adversarial fine-tuning techniques for multimodal and large foundation models [94, 95, 96]. These developments aim to address the scalability, adaptability, and cross-domain generalization challenges that limit classical adversarial training.

Despite its success, adversarial training epitomizes the ongoing cat-and-mouse dynamics of adversarial learning, each improvement in defense objectives tends to inspire new adaptive attacks that exploit implicit biases or norm constraints. Furthermore, models may exhibit robustness overfitting, maintaining stability against known attacks but failing under unseen perturbations. Consequently, emerging research emphasizes scalable and adaptive robust training frameworks that adjust perturbation strengths dynamically, incorporate certified guarantees, and jointly optimize for robustness, privacy, and fairness to ensure more durable protection in real-world systems.

5.1.3. Certified Defenses

Certified defenses aim to provide formal mathematical guarantees that a model's prediction will remain stable under bounded adversarial perturbations. One of the most widely adopted approaches is randomized smoothing [73], which constructs a smoothed classifier by averaging predictions over Gaussian noised variants of each input. This yields probabilistic certification within an ℓ_2 neighborhood, ensuring that no perturbation within that bound can alter the model's output with high confidence.

Beyond randomized smoothing, a range of analytical frameworks provide deterministic certification. Interval bound propagation computes activation bounds layer by layer, while linear relaxation and convex outer approximation methods [74] characterize decision boundaries through tractable constraints. These approaches deliver strong theoretical guarantees but remain difficult to scale to deep architectures and high dimensional data. Furthermore, certified bounds may not always capture practical attack settings, leaving residual vulnerabilities exploitable by unconstrained or distribution shifting adversaries.

The long term effectiveness of certified defenses depends on bridging this gap between theoretical robustness and practical deployment. Current efforts explore combining certification with adversarial training, leveraging mixed precision arithmetic, and incorporating probabilistic priors to reduce computational cost. In the broader adversarial landscape, certified methods serve as an essential benchmark for verifiable robustness, offering stability and trustworthiness where empirical defenses remain susceptible to the evolving arms race between attackers and defenders.

5.2. Detection Methods

Detection based defenses aim to identify adversarial inputs at inference time before they can influence model predictions. Unlike preventive or reactive mechanisms that modify training or model structure, detection methods function as monitoring layers that differentiate adversarial examples from benign data based on their statistical, neural, or behavioral inconsistencies. They are often attractive because of their modularity and ease of deployment. However, detection alone rarely provides permanent robustness, as adaptive attackers can modify perturbations to evade detection thresholds. The following subsections discuss major detection paradigms, statistical detection, neural network based detection, and behavioral analysis, while analyzing their effectiveness, limitations, and role in the ongoing cat and mouse evolution between attackers and defenders.

5.2.1. Statistical Detection

Statistical detection identifies adversarial inputs by exploiting measurable deviations from the distribution of natural data. These deviations are typically captured from internal feature representations or prediction confidence scores, where adversarial examples tend to occupy low density or high uncertainty regions of the feature space. Approaches such as kernel density estimation, confidence thresholding, and hypothesis testing are commonly used to measure these deviations.

Feinman et al. [75] combined kernel density estimation in the hidden layer space with Bayesian uncertainty metrics to flag anomalous samples. Subsequent works introduced measures such as the Maximum Mean Discrepancy (MMD) and Mahalanobis distance to quantify distributional shifts or to detect out of distribution inputs based on their proximity to class conditional centroids. These statistical techniques are lightweight, interpretable, and effective against simple perturbations, but they are vulnerable to adaptive attacks that explicitly minimize detection metrics during optimization. Moreover, the effectiveness of these methods depends heavily on accurate estimation of feature distributions, which can vary under domain shift or noisy real world data. Current research therefore emphasizes dynamic, self calibrating thresholds and hybrid pipelines that integrate statistical analysis with adversarial training or ensemble verification to sustain long term effectiveness.

5.2.2. Neural Network Based Detection

Neural based detection methods train auxiliary classifiers to distinguish adversarial inputs from benign ones by learning discriminative patterns in raw data, internal activations, or gradient features. Feature Squeezing [76] is an early example that tests a model's stability under transformations such as input quantization or smoothing. Adversarial examples often cause inconsistent predictions under these operations, whereas benign inputs remain stable. Later approaches embed detection subnetworks directly into the model architecture, using contrastive or adversarial training objectives to enhance sensitivity to malicious perturbations.

More recent designs employ ensembles of detector heads distributed across intermediate layers, combining gradient statistics, logits, and feature maps to provide a multi view defense perspective. While these methods improve detection rates, they are still susceptible to adaptive attacks that jointly optimize for both classification accuracy and detector evasion. This ongoing adversarial adaptation cycle reflects the inherent cat and mouse nature of AML: each new detection scheme motivates counter strategies that exploit its learned decision boundary. As a result, modern work increasingly focuses on hybrid detectors that combine adversarial signal amplification with certified robustness measures or meta learning based self adaptation to maintain resilience over time.

5.2.3. Behavioral Analysis

Behavioral analysis detects adversarial inputs by observing internal network dynamics such as activation trajectories, gradient patterns, or temporal consistency in model outputs. Instead of focusing solely on static input properties, these methods assess how a model behaves across layers or over sequences of inputs. For instance, activation clustering [77] identifies targeted or poisoned inputs by detecting outliers in neuron activation distributions. Temporal analysis tracks shifts in feature statistics across consecutive inference windows, while spatial consistency checks compare representations across multiple layers to identify anomalous propagation paths.

Behavioral detection provides deeper interpretability, as it reveals how adversarial perturbations alter model decision processes. However, its robustness depends on stable baseline profiles and extensive monitoring, which can increase computational overhead. Adaptive attackers can also inject perturbations that mimic benign activation trajectories, gradually eroding detection reliability. To address these issues, recent studies explore probabilistic and meta learning based behavioral models that adjust their detection thresholds dynamically according to context and history. Such adaptive frameworks represent a promising direction toward long term, self learning defenses that co evolve with adversarial strategies rather than relying on static detection rules.

Overall, detection methods serve as an important component of multi layer defense architectures but cannot guarantee lasting protection in isolation. Their continued relevance depends on integrating adaptive learning, hybridization with other defense categories, and automated retraining pipelines that evolve alongside emerging attack patterns.

5.3. Reactive Defenses

Reactive defenses are designed to mitigate or recover from adversarial interference after it has occurred, typically during inference or post deployment. Unlike preventive strategies that aim to harden models during training, reactive methods detect, correct, or adapt to adversarial inputs at runtime. This makes them particularly valuable in dynamic or high stakes environments where new attack strategies continuously emerge. However, while reactive defenses provide adaptability and fast recovery, their long term effectiveness often depends on how quickly they can evolve alongside adversarial tactics. This section discusses three major categories: model patching, input reconstruction, and ensemble based defenses, and critically examines their strengths, limitations, and ongoing challenges in sustaining robustness against adaptive attacks.

5.3.1. Model Patching

Model patching refers to updating or modifying specific layers, parameters, or modules of a deployed model to neutralize discovered vulnerabilities without performing full retraining. A common approach is adversarial fine tuning, which retrains the model on adversarial examples encountered after deployment to reinforce decision boundaries near sensitive regions. This incremental adaptation enables rapid deployment of fixes in real world systems, making it attractive for applications such as autonomous vehicles and healthcare monitoring.

Recent research extends patching with meta learning to improve generalization and responsiveness. Gupta et al. [78] proposed synthesizing patches through meta gradients that allow quick adaptation to unseen attack types. Wang et al. [79] further introduced a meta defense framework that learns universal patching strategies across tasks and attack distributions, reducing manual intervention. Despite these advances, model patching faces inherent challenges. It primarily reacts to known attack signatures and may struggle to anticipate future threats. Repeated or uncoordinated patching can degrade generalization or induce unintended biases, similar to software patch fatigue in cybersecurity. Furthermore, the lack of standardized evaluation for patched models makes it difficult to guarantee stability under adaptive adversaries. Future research must therefore integrate continual monitoring, automated vulnerability detection, and patch verification pipelines to sustain long term robustness.

5.3.2. Input Reconstruction

Input reconstruction defenses aim to restore adversarial inputs to their clean, benign forms by projecting them onto the natural data manifold. These approaches often employ generative or reconstructive models trained on legitimate data, leveraging the assumption that adversarial perturbations distort statistical regularities that can be corrected. Defense GAN [71] exemplifies this paradigm by using a GAN trained on clean samples and optimizing for a latent representation that reproduces the input while removing adversarial artifacts. Chen et al. [80] provided an extensive review of reconstruction based defenses, encompassing denoising autoencoders, manifold regularization, sparse coding, and variational inference.

Reconstruction defenses are appealing because they can act as modular preprocessing components, strengthening existing classifiers without altering their architecture. Nevertheless, several issues limit their reliability in practice. Iterative optimization introduces inference latency, and imperfect reconstructions may distort semantic features essential for correct classification. Moreover, adaptive attackers can design perturbations that survive projection or even exploit the reconstruction model itself, rendering defenses ineffective. These shortcomings highlight the broader cat and mouse pattern of adversarial learning research: every new reconstruction scheme prompts corresponding adaptive counterattacks that exploit its inductive biases. Hence, current research trends emphasize hybrid defenses that combine reconstruction with adversarial training or detection, as well as certified approaches that provide provable guarantees on projection stability.

5.3.3. Ensemble Methods

Ensemble based defenses enhance robustness by aggregating predictions or anomaly scores from multiple diverse models or feature transformations. The underlying intuition is that adversarial examples crafted for a single model rarely transfer effectively to a heterogeneous ensemble. Tramer et al. [81] introduced ensemble adversarial training, where models are trained on both their own adversarial examples and those transferred from other ensemble members, improving cross model robustness. Similarly, Xu et al. [76] developed feature squeezing, which applies transformations such as bit depth reduction or spatial smoothing to reveal inconsistencies in model predictions, enabling the detection or rejection of adversarial inputs.

While ensemble defenses provide redundancy and improve decision diversity, they also introduce practical challenges. Maintaining multiple models increases computational, memory, and energy overhead, making large ensembles infeasible for latency sensitive or resource constrained systems. Moreover, adaptive attackers can exploit shared vulnerabilities or target the ensemble aggregation logic directly, producing adversarial examples that simultaneously degrade all constituent models. Recent studies thus explore compact or hierarchical ensembles that preserve diversity with reduced cost, and ensemble distillation techniques that compress collective robustness into a single student model. Despite these advances, ensemble defenses still face the persistent challenge of sustainability. Without continual diversification and adaptive retraining, their effectiveness can erode over time as new attack strategies emerge. Nonetheless, they remain one of the most resilient and widely adopted paradigms for real world defense deployment, balancing immediate protection with the flexibility to evolve.

5.4. Privacy-Preserving Techniques

Privacy-preserving techniques are essential to safeguard sensitive user data against inference and extraction attacks in machine learning. These approaches aim to ensure that model training and inference do not reveal specific information about any individual data point. This section discusses differential privacy, federated learning, and secure aggregation, three foundational strategies used to achieve privacy guarantees in modern ML systems. While these techniques offer strong theoretical protection, their real-world deployment continues to face challenges related to efficiency, robustness, and resilience against adaptive adversaries.

5.4.1. Differential Privacy

Differential Privacy (DP) provides a rigorous mathematical framework for protecting individual data contributions in statistical computations. A mechanism is said to be differentially private if its output distribution is nearly indistinguishable when a single individual's data is modified, ensuring plausible deniability for all users whose data is involved in training. A widely used instantiation is DP-SGD [82], which modifies stochastic gradient descent by clipping gradient norms and adding calibrated Gaussian noise to each update. This process bounds the influence of any single data point on the learned model.

Foundational work by Dwork et al. [13] established the theoretical underpinnings of DP and introduced composition theorems, sensitivity analysis, and privacy budget accounting. Despite these guarantees, DP often incurs significant utility loss, particularly on small or imbalanced datasets, and its noise injection can reduce model robustness against adversarial perturbations. Moreover, adaptive adversaries can exploit cumulative gradient leakage or correlated updates to infer private attributes despite DP protection. This ongoing tension between privacy strength, model accuracy, and adversarial resilience exemplifies the cat-and-mouse dynamics in designing sustainable privacy defenses.

Differential privacy provides a formal guarantee that the output of a learning algorithm does not reveal sensitive information about any individual training sample. A randomized algorithm $\mathcal{A}$ is said to satisfy (ε, δ)-differential privacy if, for any two neighboring datasets D and D' that differ by a single record and for any measurable output set S, it holds that

$$\Pr[\mathcal{A}(D) \in S] \le e^{\varepsilon} \Pr[\mathcal{A}(D') \in S] + \delta. \tag{12}$$

In deep learning, differential privacy is commonly enforced during training via differentially private stochastic gradient descent (DP-SGD). At each iteration, per-sample gradients are clipped to a fixed norm bound C, and calibrated noise is added before aggregation:

$$\tilde{g} = \frac{1}{|B|} \left(\sum_{i \in B} \mathrm{clip}(\nabla_\theta \mathcal{L}(f_\theta(x_i), y_i), C) \right) + \mathcal{N}(0, \sigma^2 C^2 I), \tag{13}$$

where B denotes the mini-batch, σ controls the noise magnitude, and $\mathcal{N}(0, \cdot)$ is Gaussian noise.

This formulation illustrates how DP mitigates privacy attacks such as membership inference and gradient leakage by limiting the influence of any single data point on the learned model, albeit often at the cost of reduced utility or slower convergence.

5.4.2. Federated Learning

Federated Learning (FL) is a decentralized learning paradigm where multiple clients collaboratively train a global model without directly sharing their local data. Each client computes local updates and sends them to a central server, which aggregates them to update the global model. This setup reduces direct data exposure and supports compliance with privacy regulations. Bonawitz et al. [83] proposed a scalable and fault-tolerant architecture for practical FL, addressing challenges such as client dropout, straggler mitigation, and secure aggregation.

Kairouz et al. [84] provided a comprehensive overview of advances in FL, including personalization, communication efficiency, and system heterogeneity. However, FL remains vulnerable to information leakage through shared gradients, model inversion, and poisoning attacks from compromised clients. Robust aggregation and differential privacy are frequently combined with FL to mitigate these risks, though this often introduces trade-offs in model convergence and accuracy. The field continues to evolve in response to adversarial innovations, illustrating that federated learning, while privacy-enhancing, is not inherently robust without adaptive defenses and continuous verification.

Federated learning aims to train a global model collaboratively across multiple clients while keeping raw data local. Let

$\mathcal{D}_k$ denote the local dataset of client k, and let f_θ be the global model. The overall training objective can be written as

$$\min_\theta \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathbb{E}_{(x,y)\sim\mathcal{D}_k}[\mathcal{L}(f_\theta(x), y)], \tag{14}$$

where K is the number of participating clients.

In each communication round, clients compute local updates (e.g., gradients or model weights) based on their private data and send them to a central server. The server aggregates these updates using methods such as Federated Averaging (FedAvg):

$$\theta^{(t+1)} = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \theta_k^{(t)}, \tag{15}$$

where $\theta_k^{(t)}$ denotes the locally updated model at client k.

While federated learning reduces direct data exposure, this formulation highlights that shared updates can still leak sensitive information, motivating the integration of privacy-enhancing techniques such as differential privacy and secure aggregation.

5.4.3. Secure Aggregation

Secure aggregation protocols aim to protect the confidentiality of individual model updates during aggregation in federated learning settings. The central idea is that the server can compute an aggregate (e.g., sum or average) of client updates without learning any individual contribution. Hardy et al. [85] present an efficient cryptographic protocol for secure aggregation based on additively homomorphic secret sharing. Their approach is robust against client dropout and scales to large populations, making it suitable for deployment in real-world distributed systems.

Bonawitz et al. [83] integrated secure aggregation into their federated learning framework, combining it with system-level optimizations to minimize communication and computation overhead. Nevertheless, secure aggregation alone cannot prevent gradient-based inference or collusion among malicious clients. Its computational cost and dependency on strong cryptographic assumptions further limit its scalability in resource-constrained environments. Future research aims to design lightweight, verifiable aggregation schemes that maintain confidentiality while supporting adaptive threat detection, reinforcing long-term resilience against adversarial evolution.

Secure aggregation aims to prevent the server from observing individual client updates in federated learning, while still enabling correct computation of the aggregated result. Let u_k denote the local update (e.g., gradient or model difference) computed by client k. Instead of sending u_k in plaintext, each client transmits an encrypted or masked version $\tilde{u}_k$ such that the server can only recover the sum

$$\sum_{k=1}^{K} u_k, \tag{16}$$

but learns nothing about any individual update u_k.

A common approach is to use additive masking, where each client constructs

$$\tilde{u}_k = u_k + \sum_{j\neq k} r_{k,j}, \tag{17}$$

with random masks $r_{k,j}$ that cancel out when all masked updates are summed at the server. As a result, the server obtains the correct aggregated update while individual contributions remain hidden.

By ensuring that only aggregated information is revealed, secure aggregation significantly reduces the risk of gradient leakage and reconstruction attacks, especially when combined with differential privacy in federated learning systems.

5.5. Comparative Discussion and Deployment Insights

Building upon the taxonomy in Table 5, this subsection provides a comparative discussion of major defense techniques in each category, focusing on their strengths, limitations, and practical deployment challenges. While these defenses collectively enhance system robustness, each carries distinct trade-offs in terms of computational cost, adaptability, and real-world applicability.

5.5.1. Preventive Defenses

Preventive defenses aim to proactively strengthen model robustness before adversarial interference occurs. Among them, (1) *Input sanitization* techniques are lightweight and easily deployable, offering first-line protection by filtering adversarial noise through transformations such as JPEG compression or learned projections. They are effective against small perturbations but often degrade clean accuracy and can be bypassed by adaptive attacks that manipulate low-frequency or semantic features. (2) *Robust training* methods, such as adversarial and PGD-based training [17], remain the most empirically reliable strategy for in-distribution robustness. However, their computational demands and reduced clean-data accuracy limit scalability in production systems. Techniques like TRADES [72] mitigate this trade-off by explicitly balancing accuracy and robustness, though the balance remains dataset-dependent. (3) *Certified defenses* (e.g., randomized smoothing [73], convex relaxations [74]) provide provable robustness guarantees. These methods are conceptually appealing for safety-critical settings such as autonomous driving or finance, but their conservativeness and computational cost hinder widespread adoption. In practice, certified defenses are best combined with robust training to offer both empirical and theoretical resilience.

5.5.2. Detection Mechanisms

Detection-based defenses act as gatekeepers at inference time. (1) *Statistical detection* approaches are efficient and interpretable, detecting adversarial inputs by identifying deviations in confidence scores or feature distributions [75]. They are easy to integrate into deployed systems but are sensitive to threshold calibration and dataset drift, which can lead to false positives. (2) *Neural network-based detectors* leverage trainable discriminators or auxiliary models to identify adversarial patterns [76].

Table 6: Comparative Trade-offs Across Defense Techniques

Defense Category	Representative Techniques	Robustness	Accuracy Loss	Computation Cost	Deployability
Preventive	Sanitization, Robust Training, Certification	High	Medium-High	High	Moderate
Detection	Statistical, NN-based, Behavioral	Medium	Low	Low-Moderate	High
Reactive	Patching, Reconstruction, Ensemble	Medium-High	Low	Moderate-High	Moderate
Privacy-Preserving	DP, FL, Secure Aggregation	Medium (privacy)	Medium	Moderate-High	High

Table 7: Representative robust accuracy of state-of-the-art defenses discussed in this survey on standard benchmarks. The table compares clean and adversarial (PGD-10) accuracy, highlighting the trade-off between model performance and robustness.

Defense Method	Reference	Dataset	Clean Accuracy (%)	Robust Accuracy (% under PGD-10)
Adversarial Training	Madry et al. [17]	CIFAR-10	87.3	45.8
TRADES	Zhang et al. [72]	CIFAR-10	84.9	56.6
Randomized Smoothing	Cohen et al. [73]	ImageNet	69.0	43.0
Certified Defense (Linear Relaxation)	Wong et al. [74]	CIFAR-10	79.8	36.5
RobustBench Leaderboard (2025 snapshot)	Croce et al. [33]	CIFAR-10	90.2	63.0

These offer flexibility and adaptability but require diverse adversarial examples for training and may overfit to known attacks, leaving them vulnerable to unseen perturbations. (3) *Behavioral analysis* defenses monitor internal dynamics such as activation distributions and temporal variations [77]. While harder to spoof and useful in adaptive environments, they demand additional telemetry and increase inference latency. In deployment, combining statistical and behavioral detection can achieve a better balance between accuracy, interpretability, and robustness.

5.5.3. Reactive Defenses

Reactive defenses mitigate or adapt to adversarial attacks during inference or post-deployment. (1) *Model patching* [78, 79] allows targeted fine-tuning or reprogramming of vulnerable model components, offering fast mitigation with minimal retraining. Its flexibility makes it ideal for environments facing evolving threats, though patches may overfit or interfere with normal behavior if not validated carefully. (2) *Input reconstruction* approaches [71, 80] restore perturbed inputs through denoising autoencoders or generative projection. These are modular and compatible with legacy models but can introduce latency due to iterative optimization. Their success heavily depends on reconstruction quality and the representativeness of training data. (3) *Ensemble methods* [81, 76] aggregate predictions from multiple models or transformations to dilute the effectiveness of single-model attacks. They improve resilience to transferability but require additional memory and computation. In practice, ensemble diversity (different architectures, training seeds, or data subsets) is key to ensuring robustness.

5.5.4. Privacy-Preserving Techniques

Privacy-preserving mechanisms secure sensitive information against inference and extraction threats rather than direct adversarial perturbations. (1) *Differential privacy* [82, 13] provides formal guarantees by injecting calibrated noise into gradients or outputs, ensuring individual-level confidentiality. However, excessive noise can degrade accuracy, making privacy-utility trade-offs central to practical adoption. Adaptive clipping and layer-specific noise scaling are effective mitigations in large models. (2) *Federated learning* [83, 84] decentralizes training, enabling collaboration across clients without raw data exchange. It mitigates data exposure risks but remains susceptible to poisoning and gradient leakage. Integrating federated training with differential privacy and secure aggregation protocols improves robustness but increases communication and computation costs. (3) *Secure aggregation* [85, 83] employs cryptographic methods to protect individual client updates during aggregation. While it prevents information leakage from updates, it does not defend against malicious clients or model poisoning by itself. Practical deployments often pair secure aggregation with Byzantine-robust aggregation and anomaly detection to maintain both privacy and integrity in distributed learning systems. Emerging research has also explored the dual use of adversarial attacks for privacy protection, leveraging perturbations to obfuscate personal data and prevent unauthorized model inversion or extraction [97, 98, 99]. These works highlight the growing convergence between adversarial robustness and data privacy as complementary objectives in trustworthy machine learning.

5.5.5. Cross-Category Trade-offs

Table 6 summarizes the qualitative trade-offs among the four major defense categories, while Table 7 complements this view by providing quantitative benchmarks that capture the practical robustness-performance balance achieved by representative methods. Together, these tables highlight the diverse strengths and limitations across preventive, detection, reactive, and privacy-preserving strategies.

Preventive defenses, such as Adversarial Training and TRADES, continue to offer the strongest protection against perturbation-based attacks but incur nontrivial computational overhead and a reduction in clean-data accuracy, as reflected in Table 7. Detection and reactive defenses, in contrast, are more adaptable and computationally efficient, making them attractive for real-time or resource-constrained environments, though their effectiveness may degrade under adaptive adversaries. Privacy-preserving approaches like differential privacy and secure aggregation address orthogonal concerns by mitigating information leakage, yet their integration with robustness objectives often introduces additional performance trade-offs.

Quantitative results from Table 7 further reinforce these insights: while certified defenses such as randomized smoothing and linear-relaxation methods provide formal robustness guarantees, they typically achieve lower accuracy and scalability than empirical methods. In practice, hybrid frameworks that combine preventive robustness with lightweight detection and privacy-preserving components provide the most balanced defense strategy, achieving meaningful robustness without sacrificing system performance, scalability, or usability in real-world machine learning deployments.

6. Real-world Applications and Case Studies

Adversarial machine learning has progressed from a theoretical concern to a practical threat with tangible implications across numerous domains. This section highlights how AML manifests in real-world applications, affecting sectors such as computer vision, natural language processing, autonomous systems, and healthcare. As summarized in Table 8, we present case studies and industry deployments that showcase both successful adversarial attacks and the corresponding defense strategies, emphasizing the urgency and importance of addressing AML in operational settings.

6.1. Computer Vision

Computer vision systems are among the most susceptible to adversarial attacks, as they often rely on high-dimensional pixel data where imperceptible perturbations can drastically alter model predictions. Szegedy et al. [1] were among the first to demonstrate that deep convolutional neural networks could be misled by carefully crafted, low-magnitude perturbations that are visually indistinguishable to humans.

In safety-critical applications like autonomous driving, the consequences can be severe. Eykholt et al. [4] introduced a physical attack by placing small stickers on stop signs, leading state-of-the-art traffic sign classifiers to misinterpret them as speed limit signs. Adversarial patches on vehicles or pedestrians have been shown to bypass object detectors used in real-world systems such as those deployed by Tesla and Waymo, raising concerns about road safety and regulatory compliance.

Face recognition and surveillance systems are also prime targets. Attackers have developed adversarial accessories, such as eyeglass frames or patterned masks, that allow individuals to evade detection or impersonate others in facial authentication frameworks. In commercial applications like retail analytics and smart city infrastructure, adversarial perturbations can disrupt people-counting systems, gesture recognition interfaces, and behavioral analytics pipelines, leading to both privacy violations and operational failures.

To counter these threats, a range of defenses has been explored. Adversarial training remains the most widely used technique, although it is computationally expensive and task-specific. Input preprocessing methods, such as JPEG compression [70], image quilting, and bit-depth reduction, have shown some efficacy in removing adversarial noise. Generative approaches like Defense-GAN [71] attempt to project inputs back onto the data manifold. Ensemble-based strategies that combine predictions from multiple independently trained models also improve resilience. In industrial deployments, multi-sensor fusion, cross-view consistency checks, and hardware redundancy are increasingly being used to bolster robustness against visual adversarial attacks.

6.2. Natural Language Processing

In natural language processing, adversarial attacks exploit the inherent flexibility of human language (e.g., semantic equivalence, syntactic variation, and character-level mutations) to deceive models. Unlike image perturbations, text-based adversarial examples must remain grammatically correct and semantically plausible to avoid human detection, making attack generation both challenging and impactful.

Ebrahimi et al. [100] introduced HotFlip, a white-box attack that uses gradients to identify optimal character-level changes capable of altering sentiment predictions. These minimal edits, such as character swaps or deletions, can lead to misclassifications in tasks like spam detection, hate speech filtering, and legal document classification. More recent attacks target word substitutions using embedding-based similarity metrics to maintain fluency while degrading model performance.

Beyond text, adversarial attacks have also emerged in the audio modality of NLP. Carlini and Wagner [101] crafted audio adversarial examples that embed hidden voice commands. These imperceptible perturbations activate smart assistants like Alexa or Siri, exploiting the model's sensitivity to phonetic ambiguity and background noise. Such attacks pose serious risks to privacy and device control.

Defensive techniques in NLP include embedding-stable training that encourages consistency across semantically similar inputs, and adversarial data augmentation using paraphrases or syntactic variants [102]. Adversarial regularization penalizes high sensitivity to small textual changes during training, while runtime defenses such as spell-checkers or grammar-aware filters can mitigate some character-level attacks. Models fine-tuned on adversarially augmented datasets, particularly large transformer models like BERT, exhibit improved robustness in production systems deployed in content moderation, sentiment analysis, and voice-based interfaces.

6.3. Autonomous Systems

Autonomous platforms, such as self-driving vehicles, drones, and robotic systems, rely heavily on machine learning for tasks including perception, localization, planning, and control. This tight integration makes them prime targets for adversarial attacks. Cao et al. [103] demonstrated that adversarial perturbations on LiDAR point clouds can either fabricate phantom

Table 8: Summary of Real-World Applications of Adversarial Machine Learning

Domain	Example Attacks	Real-World Defenses
Computer Vision	Image misclassification, adversarial patches, face evasion attacks	Adversarial training, input preprocessing, model ensembles [1, 4, 70]
NLP	Synonym substitution, perturbations, adversarial audio	Paraphrase-invariant training, adversarial BERT tuning [100, 101, 102]
Autonomous Systems	Road sign spoofing, LiDAR hallucination, navigation error	Sensor fusion, consistency checks, secure firmware [103, 104]
Healthcare	Misdiagnosed medical images, ECG waveform manipulation	Certified imaging pipelines, denoising autoencoders [5, 105]
Finance	Fraud pattern evasion, adversarial credit scoring, market spoofing	Ensemble anomaly detection, adversarial retraining [106, 107]
Network Traffic	Adversarial detection, GAN-based attacks	Strengthen feature representation, obfuscation [108, 109, 110, 111]
Cybersecurity	Malware mutation, adversarial traffic patterns	Byte masking, adversarial honeypots [112, 113]
Content Moderation	Toxic content evasion, adversarial paraphrasing	Graph-based filters, hybrid human-AI moderation [114, 115]
Industrial Systems	Sensor spoofing, smart grid attacks	Secure enclave control, protocol verification [116, 117]

objects or occlude real obstacles, severely impairing 3D object detection systems used in autonomous driving.

In industrial automation, adversarial floor patterns have been shown to confuse vision-based navigation systems in warehouse robots and automated guided vehicles (AGVs), leading to navigation errors or system halts. Aerial systems are similarly susceptible, adversarial camouflage techniques can reduce the effectiveness of onboard object detection in drones, impacting surveillance and reconnaissance accuracy. These threats have serious implications for both civilian and military autonomous systems, where reliability is mission-critical.

In response, industry leaders such as NVIDIA, Boston Dynamics, and Cruise have incorporated defenses such as sensor fusion (e.g., combining camera, radar, and LiDAR), temporal consistency checks, and adversarially retrained models into their ML pipelines. Moreover, government-backed efforts like DARPA's Assured Autonomy initiative [104] focus on developing formal verification frameworks to ensure safety guarantees even under adversarial conditions. These efforts underscore the urgent need for robust and certifiable autonomy in adversarial environments.

6.4. Healthcare

Adversarial machine learning in healthcare carries potentially life-threatening consequences due to the high stakes involved in clinical decision-making. Finlayson et al. [5] demonstrated that minor perturbations to dermoscopic images could lead to misclassification by dermatology AI systems, mistaking benign lesions for malignant ones and vice versa. Similarly, Hannun et al. [105] showed that adversarial noise introduced into ECG signals could disrupt arrhythmia classification, risking misdiagnosis or treatment delay.

Beyond image and signal data, attacks on electronic health record (EHR) systems and AI-based clinical decision support tools have shown that adversarial manipulation of structured patient data can lead to incorrect treatment recommendations. Natural language generation models used in radiology reporting or diagnostic summarization can also be manipulated with token-level perturbations that subtly alter medical conclusions.

Emerging threats include signal injection attacks on wearable medical devices such as continuous glucose monitors, heart-rate trackers, and insulin pumps, which may exploit wireless vulnerabilities or physiological signal spoofing.

To counteract these threats, medical device manufacturers like Siemens and Philips are adopting defenses including secure enclaves, encrypted ML inference pipelines, adversarial input denoising, and formal model auditing. Furthermore, the U.S. Food and Drug Administration (FDA) is increasingly mandating adversarial robustness evaluations as part of its regulatory approval process for AI-based medical systems, signaling a shift toward proactive security in digital healthcare.

6.5. Finance and Fraud Detection

Adversarial machine learning presents significant risks in financial systems where models are used for fraud detection, credit scoring, transaction monitoring, and algorithmic trading. Attackers can manipulate transaction logs or behavioral sequences to evade detection. For instance, adversaries may split large fraudulent transactions into smaller amounts, structure spending patterns to resemble legitimate users, or subtly alter features like transaction time, location, or merchant type to exploit model blind spots [106, 107].

Credit scoring systems are also vulnerable to feature manipulation attacks. By strategically modifying input variables such as income level, credit utilization, or address history, attackers can receive inflated credit ratings without altering their actual risk profile. Such manipulations can be executed through adversarial optimization algorithms designed to remain within acceptable feature bounds, making them hard to detect during auditing or verification.

High-frequency and algorithmic trading platforms are particularly sensitive to real-time data inputs. Malicious actors can introduce synthetic patterns or noise into data feeds to manipulate the behavior of automated trading bots, potentially triggering flash crashes or market instability. These adversarial signals may exploit vulnerabilities in prediction models used for price movement forecasting or trade volume estimation.

In response, financial institutions such as Mastercard, PayPal, and Capital One have implemented ensemble-based fraud detection systems that leverage diverse classifiers trained on different data slices or detection objectives. These ensembles are often accompanied by robust logging mechanisms and explainability modules to facilitate post-incident forensics. Adversarial training, model confidence calibration, and input validation are also incorporated into production pipelines to harden models against subtle evasion techniques and maintain robustness over time. Moreover, regulatory frameworks are increasingly recognizing the need for adversarial robustness as a component of AI model governance and auditability in the financial sector.

6.6. *Network Traffic*

Adversarial machine learning has increasingly extended into the domain of network and encrypted-traffic security, where deep learning models are extensively employed for intrusion detection, service identification, and traffic classification. While these models have achieved high performance by leveraging statistical and temporal flow characteristics, their dependence on such implicit patterns also introduces new adversarial vulnerabilities, particularly in encrypted communication scenarios where payload inspection is infeasible.

Recent studies have explored this emerging area from both offensive and defensive perspectives. The CBS framework [108] presents a deep learning architecture that integrates spatial, temporal, and statistical representations to classify encrypted traffic, demonstrating the potential of multi-feature fusion for more resilient network modeling. Similarly, AD-VoIP [109] investigates adversarial detection of encrypted and concealed VoIP flows, showing how AML methodologies can be applied to uncover hidden communication behaviors under encryption and obfuscation.

Complementary research has analyzed vulnerabilities and proposed countermeasures in encrypted-traffic analytics. Liu et al. [110] examined adversarial obfuscation techniques that modify encrypted packet sequences to evade traffic classification systems, exposing critical weaknesses in existing deep models. More recently, Zhan et al. [111] introduced the EAPT framework, which leverages adversarial pre-training based on transformer architectures to strengthen feature representation and enhance robustness against adversarial perturbations in encrypted traffic classification.

Collectively, these studies highlight the growing importance of adversarial learning in modern network security. As traffic increasingly shifts toward encrypted and privacy-preserving protocols, robust AML-based techniques that can maintain accurate classification, preserve user privacy, and withstand adversarial manipulation will be essential for securing next-generation communication infrastructures.

6.7. *Cybersecurity and Intrusion Detection*

Machine learning models deployed in cybersecurity, such as malware classifiers and intrusion detection systems (IDS), are attractive targets for adversarial attacks. These systems often analyze high-dimensional, structured inputs like byte sequences or network flows, which adversaries can subtly perturb to evade detection. Hu and Tan [112] demonstrated that adversarial traffic traces can be crafted to mimic benign behavior, allowing attackers to bypass IDS without raising alerts.

Grosse et al. [113] further exposed the vulnerability of DNN-based malware classifiers to adversarial byte-level modifications that retain malware functionality while flipping predictions. Attackers also exploit adversarial evasion techniques in command-and-control traffic, phishing payloads, and polymorphic malware, complicating traditional signature-based and behavioral detection.

To counter these threats, security practitioners have begun integrating byte-level masking, adversarial training, and anomaly detection using attention-based models. Deception-based defenses, such as honeypots enhanced with adversarial awareness, lure attackers while collecting training data to improve detection robustness. Toolkits like IBM's Adversarial Robustness Toolbox (ART) and Microsoft's Counterfit facilitate automated robustness evaluation. Nonetheless, balancing detection sensitivity and false positive rates under adversarial pressure remains a critical challenge in operational environments.

6.8. *Content Moderation and Recommender Systems*

Content moderation and recommender systems, which rely heavily on natural language processing and user engagement signals, are susceptible to adversarial manipulation. Attackers can construct fake profiles, generate synthetic reviews, or manipulate click-through data to promote harmful content or demote legitimate items. These attacks distort model behavior in applications like product recommendations, news feeds, and content curation.

Moderation systems are similarly vulnerable. Adversaries exploit semantic-preserving transformations, obfuscations, and typographic attacks (e.g., homoglyphs or zero-width characters) to evade hate speech, spam, and misinformation filters [114, 115]. These evasive inputs challenge even state-of-the-art toxicity classifiers deployed in social media and e-commerce platforms.

To mitigate these risks, companies like YouTube, TikTok, Facebook, and Reddit utilize hybrid moderation pipelines. These systems combine adversarially trained classifiers, rule-based filters, and manual reviews. Graph-based anomaly detection helps identify coordinated inauthentic behavior, including botnets and adversarial influence operations. In recommender systems, countermeasures such as robust matrix factorization, user credibility scoring, and diversity-aware ranking are adopted to reduce the impact of adversarial manipulation. Continued advancements in model interpretability and real-time monitoring are essential for securing these systems.

6.9. *Industrial Control Systems*

Industrial control systems (ICS), including those in manufacturing, utilities, transportation, and energy, are increasingly reliant on AI for predictive maintenance, anomaly detection, and process optimization. These systems are particularly vulnerable to adversarial attacks due to the physical consequences of erroneous predictions. Zhang et al. [117] demonstrated that adversarial perturbations on smart grid demand forecasting could lead to large-scale instability, resulting in inefficient resource distribution or service disruptions.

Adversaries can also inject false sensor readings into programmable logic controllers (PLCs) or supervisory control and data acquisition (SCADA) systems, causing actuators to misbehave. For example, modifying temperature or vibration signals can prevent timely maintenance alerts, while concealing pipeline leaks or overheating events. Such attacks can be stealthy, persistent, and hard to detect without physical redundancy or secure sensing infrastructure.

Defensive strategies for ICS include physics-informed neural networks that embed domain knowledge into model constraints, certified control policies that enforce robustness guarantees under bounded perturbations, and secure enclave deployment (e.g., Intel SGX) to prevent tampering during inference [116]. Industrial players like Siemens, ABB, and Honeywell are increasingly investing in secure AI platforms and collaborating with national agencies. Regulatory bodies, including the U.S. NIST and the European ENISA, are developing guidelines and certification processes to promote the resilience of industrial AI systems under adversarial conditions.

6.10. Robustness Benchmarks and Evaluation Frameworks

Beyond domain-specific case studies, the assessment of adversarial robustness increasingly relies on standardized benchmarking frameworks that provide reproducible and comparable evaluations. Two prominent examples are *RobustBench* and *AutoAttack*, which have become widely adopted in both academic and industrial settings.

RobustBench [33] serves as an open benchmark for evaluating and tracking the progress of robust models under standardized threat models. It maintains a public leaderboard covering a variety of datasets such as CIFAR-10, CIFAR-100, and ImageNet, enabling consistent comparisons of defenses across architectures and attack methods. RobustBench emphasizes transparency, encouraging the community to submit verified robustness scores obtained using unified testing pipelines.

AutoAttack [118] provides a strong, parameter-free ensemble of adversarial attacks that evaluate model robustness in a deterministic and reproducible manner. It eliminates the tuning bias present in earlier attack evaluations, ensuring fairness and reliability. AutoAttack has become a de facto standard for robustness verification in industrial pipelines, where reproducibility and efficiency are essential.

The integration of such benchmarks bridges the gap between research and deployment by promoting standardized robustness metrics. These frameworks also assist practitioners in model certification, compliance auditing, and performance tracking under adversarial conditions. As a result, they play a crucial role in transitioning adversarial defense techniques from controlled laboratory environments to production-grade, real-world systems.

7. Observations and Lessons Learned

Through the comprehensive review of adversarial attacks, defense mechanisms, and real-world applications presented in this paper, several key observations and lessons emerge. These insights reveal the evolving nature of adversarial machine learning and underscore the theoretical, practical, and systemic factors that shape progress toward robust and trustworthy AI systems.

7.1. Dynamic and Systemic Nature of Adversarial Robustness

Adversarial robustness is inherently dynamic rather than static. Each generation of defenses, such as adversarial training, certified robustness, and privacy-preserving learning, has eventually been overcome by newly adapted attack strategies. This continuing evolution highlights that robustness is not a permanent property of a model but an ongoing process that must be sustained through continuous retraining, adaptive evaluation, and evolving threat modeling. Furthermore, robustness is not confined to model design alone but extends across the entire machine learning lifecycle. Many successful attacks, such as data poisoning and backdoor insertion, exploit vulnerabilities in data collection, labeling, or retraining pipelines. Therefore, achieving robustness requires a holistic, system-level approach that integrates data integrity assurance, secure model updates, and proactive monitoring. Robustness must evolve alongside both technological progress and adversarial sophistication.

7.2. Trade-offs, Evaluation Practices, and Domain Constraints

Another central lesson is that adversarial robustness always entails trade-offs among accuracy, efficiency, interpretability, and deployability. Preventive defenses, such as adversarial training, often achieve strong robustness but incur high computational costs and degrade clean-data accuracy. Detection and reactive methods are more lightweight but frequently attack-specific and susceptible to adaptive evasion. Privacy-preserving defenses mitigate information leakage but can amplify learning noise and reduce predictive utility. The suitability of a defense thus depends on domain-specific requirements such as latency, resource constraints, and safety assurance. In mission-critical fields like autonomous driving, healthcare, and finance, these trade-offs determine whether a defense is viable in practice. The emergence of standardized robustness benchmarks, including *RobustBench* and *AutoAttack*, represents an important step toward reproducible and transparent evaluation. Nevertheless, future evaluation frameworks must account for adaptive threat models, domain-specific constraints, and realistic deployment scenarios to ensure the credibility of robustness claims.

7.3. Best Practices for Robustness Evaluation

To ensure rigorous and reproducible evaluation of adversarial defenses, the following checklist summarizes key community-accepted best practices:

- **Adopt adaptive attack settings:** Evaluate defenses under adaptive, white-box conditions where the attacker is fully aware of the defense mechanism. Avoid gradient masking and obfuscation artifacts by verifying that attack gradients remain informative [119].

- **Benchmark with standardized frameworks:** Utilize open platforms such as *RobustBench* and *AutoAttack* for fair, comparable robustness measurement. These frameworks provide unified evaluation metrics for clean accuracy, ℓ_p-bounded robustness (e.g., PGD-10), and certified guarantees across model architectures.

- **Include clean and adversarial accuracy:** Report both standard (clean) and adversarial accuracies to highlight robustness-accuracy trade-offs and to prevent misleading claims of security-through-obfuscation.

- **Leverage verification and certification tools:** Apply formal verification methods such as randomized smoothing [73] and convex relaxation [74] to provide certified robustness bounds where applicable.

- **Ensure reproducibility:** Publish attack configurations, random seeds, and model checkpoints. Incorporate adaptive attack re-runs in supplementary materials or appendices to verify defense stability.

These practices collectively promote transparency, comparability, and scientific rigor in adversarial robustness evaluation.

7.4. Toward Integrated and Trustworthy Robustness

Finally, adversarial robustness should be pursued as part of a broader vision of trustworthy AI that also encompasses privacy, fairness, and interpretability. These aspects are deeply interconnected: for example, differential privacy can protect individual data but may inadvertently weaken robustness if noise injection is excessive, while robustness-oriented defenses can shift model behavior in ways that affect fairness or transparency. Addressing these interdependencies requires cohesive design frameworks that jointly optimize across these objectives rather than treating them as isolated problems. Future progress will depend on cross-disciplinary collaboration among researchers in security, machine learning, and systems engineering, supported by formal verification, standardized reporting, and real-world stress testing. Ultimately, lasting robustness will emerge from adaptive, scalable, and ethically grounded approaches that integrate adversarial resilience with the broader goals of reliable and trustworthy AI.

These observations and lessons lay the groundwork for the following section, which discusses open challenges and research directions that remain critical for advancing the field of adversarial machine learning.

8. Challenges and Open Problems

Despite notable progress in identifying, characterizing, and mitigating adversarial threats, adversarial machine learning continues to face a range of fundamental challenges that hinder both theoretical understanding and practical deployment. These challenges stem from the intrinsic complexity of modern machine learning systems, the adaptive and strategic nature of adversaries, and persistent limitations in existing modeling assumptions, evaluation methodologies, and data collection practices. As learning-based systems are increasingly integrated into safety-critical and privacy-sensitive domains, such as healthcare, autonomous systems, and financial services, the impact of these unresolved issues becomes more pronounced and consequential.

Many proposed attack and defense techniques remain effective only under narrowly defined threat models or controlled experimental settings, and their reliability often degrades when confronted with adaptive adversaries, distribution shifts, or real-world operational constraints. Moreover, gaps between theoretical robustness guarantees and empirical performance, as well as mismatches between benchmark datasets and deployment environments, further complicate the assessment of adversarial risk. This section focuses on systematically identifying and analyzing the key technical and practical bottlenecks that currently limit the robustness, reliability, and deployability of adversarial machine learning techniques, and highlights the major open problems that continue to challenge the field.

8.1. Technical Bottlenecks in Adversarial Robustness

A central technical bottleneck in adversarial machine learning lies in achieving robustness that generalizes across attack strategies, threat models, and deployment environments. Many existing defenses are designed under restrictive assumptions, such as bounded perturbations in specific norm spaces or static attacker capabilities. While such assumptions simplify analysis, they often fail to capture the diversity and adaptability of real-world adversaries. As a result, defenses that appear effective under controlled experimental settings frequently break down when attackers deviate from assumed threat models or exploit unforeseen vulnerabilities.

Another major technical challenge stems from the growing complexity of modern machine learning architectures. Deep neural networks with millions or billions of parameters, pretraining pipelines, and fine-tuning stages introduce additional layers of vulnerability that are difficult to analyze formally. Robustness guarantees may not transfer across datasets, tasks, or model architectures, suggesting that many defenses exploit dataset-specific artifacts rather than fundamental invariances. These issues are further amplified in multimodal and foundation models, where interactions between modalities introduce new and poorly understood attack surfaces. Overcoming these bottlenecks requires new theoretical tools and defense mechanisms that offer robustness guarantees beyond narrowly defined adversarial settings.

8.2. Methodological Limitations and Evaluation Challenges

Methodological limitations in robustness evaluation represent another major obstacle to progress in adversarial machine learning. A significant portion of the literature relies on incomplete threat models, limited attack diversity, or non-adaptive evaluation procedures. Such practices can lead to misleading conclusions about robustness, as attackers in real-world settings are often adaptive and capable of exploiting defense-specific weaknesses. Historical failures of defenses based on gradient obfuscation or input preprocessing underscore the risks of insufficient evaluation rigor.

Although recent efforts have introduced standardized benchmarks and evaluation platforms, several methodological challenges remain unresolved. There is still no consensus on how to model realistic attacker knowledge, how to balance white-box and black-box evaluations, or how to fairly compare defenses with fundamentally different assumptions and goals. Moreover, robustness metrics are often heterogeneous and task-dependent, making cross-paper comparisons difficult. Addressing these methodological issues requires the development of unified evaluation protocols, stronger adversarial testing practices, and reproducible pipelines that better reflect real-world adversarial behavior.

8.3. Data Availability and Realism Challenges

Data-related challenges pose a significant bottleneck for both adversarial attack research and defense validation. Constructing realistic adversarial datasets is inherently difficult due to the high cost of data collection and labeling, especially in domains where expert knowledge is required. Additionally, adversarial behavior is highly dynamic, making static datasets insufficient to capture evolving attack strategies. As a result, many studies rely on simplified or synthetic benchmarks that fail to reflect real-world operating conditions.

These limitations are further compounded by privacy, legal, and ethical constraints on data sharing. In sensitive application domains such as healthcare, finance, and critical infrastructure, access to representative datasets is often restricted, limiting empirical evaluation of adversarial threats. This gap between benchmark datasets and deployment environments can lead to defenses that generalize poorly in practice. Addressing these challenges requires new approaches to data generation, simulation, and sharing that balance realism, privacy preservation, and reproducibility, as well as domain-aware evaluation methodologies.

8.4. Practical Deployment Constraints

Beyond algorithmic considerations, practical deployment introduces constraints that are frequently overlooked in adversarial machine learning research. Many state-of-the-art defenses incur substantial computational overhead, increased inference latency, or additional memory consumption. These costs may be acceptable in offline evaluation settings but become prohibitive in real-time or resource-constrained environments such as edge devices, mobile platforms, or embedded systems. Consequently, highly robust defenses may be impractical for large-scale deployment despite strong empirical performance.

Integrating adversarial defenses into existing machine learning pipelines also raises system-level challenges. Models must be maintained, updated, and monitored over time, and defenses must remain effective under distribution shifts and evolving adversarial strategies. In safety-critical and regulated domains, additional requirements such as interpretability, certification, and compliance further complicate deployment. These practical constraints highlight a persistent gap between laboratory-scale robustness research and operational security needs, underscoring the importance of designing defenses that are not only robust but also efficient, maintainable, and deployable in real-world systems.

8.5. Scalability Issues

Scaling defense mechanisms to large-scale datasets and complex models in a computationally efficient manner remains a critical challenge. Techniques such as adversarial training, particularly with strong attacks like Projected Gradient Descent (PGD) [17], significantly prolong training time and demand substantial computational resources. Similarly, certified defenses such as randomized smoothing [73] offer formal robustness guarantees but require thousands of Monte Carlo samples per prediction, rendering them impractical for high-resolution datasets like ImageNet or time-sensitive applications.

This computational burden is particularly problematic for real-world systems in domains such as autonomous driving, finance, or healthcare, where latency and energy constraints are stringent. High-overhead defenses may not be compatible with edge deployment or real-time inference requirements. As such, a key open problem is designing scalable, hardware-aware, and modular defense techniques that retain robustness without compromising throughput, energy efficiency, or model performance in production environments.

8.6. Transferability of Attacks

One of the most puzzling and dangerous characteristics of adversarial examples is their transferability across models. Papernot et al. [120] demonstrated that adversarial inputs crafted for a surrogate model can often mislead different models trained on similar or even disjoint datasets. This property enables powerful black-box attacks that require little to no access to the target model's internals, posing serious risks in settings where models are deployed as opaque APIs or proprietary services.

The underlying causes of transferability remain poorly understood. Factors such as shared decision boundaries, model overparameterization, and feature alignment are believed to contribute, but a formal characterization is lacking. Transferability also appears to vary with task modality (e.g., vision vs. NLP), data distribution, and training procedure (e.g., pretraining or fine-tuning). While ensemble-based defenses [81] can reduce transferability by promoting diversity among models, they are computationally expensive and not universally effective. Addressing this open problem requires deeper theoretical insights and the development of model architectures or training paradigms inherently resistant to cross-model attacks.

8.7. Defense Robustness

Evaluating the robustness of adversarial defenses remains an elusive and evolving challenge. Many early defenses, such as those based on gradient obfuscation or input preprocessing, were later shown to offer a false sense of security and were easily bypassed by adaptive adversaries [119]. These failures highlight the importance of strong threat models and rigorous, white-box evaluation procedures. Yet, many published defenses are only tested against limited attacks or fail to consider adaptive scenarios.

Although standardized platforms such as RobustBench have emerged to facilitate consistent comparisons, the AML community still lacks consensus on evaluation protocols that reflect real-world adversaries. A critical need exists for comprehensive benchmarks, standardized threat models, and reproducible evaluation pipelines that include adaptive attacks, black-box scenarios, and domain-specific considerations. Without these, claims of robustness risk being misleading or overly optimistic.

8.8. *Robustness vs Accuracy Tradeoff*

One of the most fundamental challenges in adversarial machine learning is the apparent tradeoff between robustness and accuracy. Techniques such as adversarial training enhance robustness by incorporating adversarial examples during training, but this often leads to reduced accuracy on clean, unperturbed data [72]. This compromise complicates deployment decisions, particularly in production systems where even marginal reductions in accuracy may be unacceptable.

This tradeoff is especially consequential in safety-critical applications such as healthcare diagnostics or autonomous navigation, where both high accuracy and strong robustness are simultaneously required. Promising directions to address this challenge include multi-objective optimization frameworks, robustness-aware regularization schemes, and hybrid models that adaptively switch between robust and accurate inference modes based on input uncertainty.

8.9. *Privacy vs Utility*

Privacy-preserving methods, particularly differential privacy (DP), offer formal guarantees against information leakage but often at the cost of model performance. Abadi et al. [82] demonstrated that while DP-SGD can limit data exposure, it also substantially reduces accuracy on standard benchmarks. This degradation arises from the injected noise and gradient clipping that are necessary to enforce privacy guarantees.

The tension between privacy and utility is especially acute in federated learning systems, where model quality, communication efficiency, and user data confidentiality must be simultaneously optimized. Addressing this challenge requires advances in adaptive noise calibration, privacy accounting techniques such as Rényi differential privacy, and utility-aware training strategies that minimize the impact of privacy constraints on overall performance. Balancing these competing goals remains an active area of research with high practical relevance.

9. Future Research Directions

To address the persistent challenges in adversarial machine learning, this section outlines promising avenues for future research. These directions span technical innovations, system-level designs, and emerging policy considerations.

9.1. *Improving Robustness in Real-World Scenarios*

While many AML defenses demonstrate efficacy in controlled academic environments, they often fail under real-world conditions. For example, defenses against physical attacks, such as perturbations to stop signs in autonomous vehicles, must remain effective under varying lighting, angles, and weather conditions [4].

Future work should prioritize domain-specific robustness tailored to high-stakes sectors like finance, healthcare, transportation, and critical infrastructure. Online learning and continual adaptation can help models dynamically respond to evolving adversarial strategies. Benchmarking efforts should also include realistic deployment scenarios with environmental noise and system constraints.

9.2. *Mitigating and Understanding Transferability*

As discussed earlier, the transferability of adversarial examples across models remains a key obstacle to security. Adversarial inputs crafted for one model often generalize to others, regardless of architectural differences or training data [120].

Future research should focus on characterizing the underlying mechanisms that facilitate transferability and developing defenses that disrupt these commonalities. Promising techniques include randomized model ensembles, adversarial feature decorrelation, and obfuscation of decision boundaries. Theoretical frameworks to quantify and bound transferability would also improve defense design.

9.3. *Privacy and Robustness Synergies*

Although robustness and privacy are often pursued independently, their objectives intersect in many real-world ML systems. For example, models that are both privacy-preserving and robust are increasingly needed in sensitive domains such as medical diagnostics and mobile applications.

Future research should develop unified frameworks that jointly optimize for both properties. For instance, combining adversarial training with differential privacy [82] remains a computationally intensive process, often degrading performance. Efficient joint training strategies, adaptive noise injection, and theoretical trade-off analysis could lead to more practical solutions.

9.4. *Scalable Defense Mechanisms*

Scalability remains a fundamental requirement for deploying AML defenses in real-world applications. Methods such as randomized smoothing [73] provide certified robustness but are computationally expensive due to extensive sampling.

To improve scalability, researchers can explore sparsity-inducing training, low-rank model compression, and lightweight defense layers amenable to hardware acceleration. In federated learning contexts [84], communication-efficient and distributed defenses are essential. Designing decentralized protocols with adversarial robustness guarantees remains a critical open problem.

9.5. Reinforcement and Federated Learning Models

Reinforcement learning (RL) systems introduce unique adversarial vulnerabilities because agents continuously interact with dynamic environments. Attackers can manipulate reward signals, observations, or transition dynamics to corrupt policy learning, leading to unsafe or suboptimal behaviors. Such manipulations, including reward shaping and policy induction, undermine the reliability of autonomous decision-making in applications like robotics and network control. Future work should focus on secure policy learning, robust environment modeling, and verification frameworks that ensure resilience against adversarial perturbations during training and deployment.

Federated learning (FL) poses distinct challenges arising from decentralized collaboration among clients with local data. While FL enhances privacy by keeping data on devices, it remains vulnerable to model poisoning, backdoor insertion, and gradient-based inference attacks. Malicious clients may submit crafted updates to bias the global model or leak private information through gradients. Emerging defenses, such as robust aggregation, anomaly detection, and secure multi-party computation, help mitigate these risks. However, achieving a balance among robustness, privacy, and communication efficiency remains an open research problem.

9.6. Foundation and Large Language Models

The emergence of foundation models, including large language and vision-language models, introduces a new adversarial landscape that differs markedly from traditional classification settings. These systems operate in open-ended, instruction-driven environments, creating novel vulnerabilities such as prompt injection, data extraction, and adversarial fine-tuning.

Prompt Injection and Instruction Manipulation. Attackers can craft malicious prompts to override alignment constraints, induce policy violations, or bypass safety filters. Such prompt-injection and jailbreak attacks exploit the model's reliance on contextual instructions rather than fixed inputs, exposing weaknesses in rule-based alignment and content moderation.

Data Extraction and Adversarial Fine-Tuning. Models trained on massive, uncurated datasets are prone to memorizing sensitive information, allowing adversaries to extract personal or proprietary data through targeted queries. Meanwhile, adversarial fine-tuning enables attackers to implant hidden behaviors or backdoors during downstream adaptation, threatening model reliability and trustworthiness.

Addressing these challenges requires combining robustness, privacy, and alignment strategies, such as differential privacy, provenance tracking, and continuous red-teaming, to build resilient foundation models. As these models increasingly underpin critical applications, adversarial robustness at this scale represents a key frontier for future AML research.

9.7. Ethical and Regulatory Frameworks

The societal implications of AML are far-reaching. As adversarial threats can influence critical systems such as medical diagnostics, transportation, and content moderation, ethical and regulatory considerations become essential. Dual-use risks, liability for mispredictions, and access equity are major concerns.

Interdisciplinary collaboration among machine learning researchers, legal scholars, ethicists, and policymakers is needed to formulate actionable guidelines. Regulatory frameworks such as the EU AI Act and U.S. FDA guidance may soon incorporate robustness requirements, underscoring the urgency of developing AML-aware compliance standards.

9.8. Integrating Adversarial Robustness with Trustworthy AI

Adversarial robustness should be studied not in isolation but as a key pillar of trustworthy AI, alongside fairness, explainability, accountability, and certification. Robust models that fail to maintain fairness across demographic groups or lack interpretability may still be unsuitable for deployment in safety- or ethics-sensitive domains. Similarly, certification frameworks for AI systems, such as robustness verification, uncertainty quantification, and transparency audits, are becoming increasingly intertwined with adversarial defense research.

Future work should focus on developing unified frameworks that jointly consider robustness and trustworthiness objectives. For instance, explainable robustness techniques could help identify which features contribute most to adversarial vulnerability, thereby improving interpretability and auditability. Fairness-aware adversarial training can mitigate bias amplification under attack conditions. Moreover, integrating robustness certification into standardized trustworthy AI benchmarks would enable regulators and practitioners to assess system reliability under both ethical and adversarial dimensions. Progress in this direction will help bridge the gap between technical defenses and societal expectations for secure, fair, and accountable AI systems.

10. Conclusions

Adversarial machine learning has emerged as a critical field at the intersection of machine learning and security, revealing fundamental vulnerabilities that challenge the integrity and trustworthiness of AI systems. Since the initial discovery of adversarial examples, the field has rapidly evolved to encompass a broad range of attack vectors, defense strategies, theoretical foundations, and practical considerations. As AI continues to permeate safety-critical domains, the need to understand and mitigate adversarial risks has become increasingly urgent.

This survey has provided a comprehensive overview of the AML, covering core attack methodologies, diverse defense mechanisms, and emerging applications across sectors such as healthcare, autonomous systems, finance, and cybersecurity. Through case studies and empirical evidence, we have illustrated the real-world impact of adversarial threats and highlighted the growing importance of deploying scalable, privacy-preserving, and robust machine learning models.

Despite significant progress, adversarial machine learning still faces open challenges, including the scalability and deployability of defenses, the persistent issue of attack transferability, and the inherent trade-offs among robustness, accuracy, and privacy. Addressing these challenges requires interdisciplinary collaboration and sustained innovation across technical, ethical, and regulatory fronts. As AI continues to expand into emerging paradigms such as reinforcement learning, federated learning, and foundation models, the development of secure, adaptive, and trustworthy learning systems becomes increasingly critical.

Looking ahead, the continued advancement of adversarial machine learning will depend on progress along several key dimensions. A major challenge lies in developing robustness that generalizes beyond narrowly defined threat models and remains effective against adaptive and unforeseen adversaries. At the same time, improving evaluation practices under realistic attack assumptions and bridging the gap between academic benchmarks and deployment environments are essential for establishing trustworthy robustness claims. Emerging directions such as adversarially resilient decentralized learning, robust reinforcement learning, and secure foundation models present both new opportunities and urgent challenges. Progress along these directions will be crucial for translating advances in adversarial machine learning from controlled research settings into reliable, secure, and trustworthy AI systems deployed in the real world.

Acknowledgement: Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-24-1-0088. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations (ICLR), 2014.

[2] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2015).

[3] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, J. D. Tygar, Adversarial machine learning, in: Proceedings of the 4th ACM workshop on Security and Artificial Intelligence, ACM, 2011, pp. 43–58.

[4] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust physical-world attacks on deep learning visual classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1625–1634.

[5] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, I. S. Kohane, Adversarial attacks on medical machine learning, Science (2019).

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

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

[8] X. Wang, H. Jiang, T. Zeng, Y. Dong, An adaptive fused domain-cycling variational generative adversarial network for machine fault diagnosis under data scarcity, Information Fusion (2025) 103616.

[9] X. Wang, H. Jiang, M. Mu, Y. Dong, A trackable multi-domain collaborative generative adversarial network for rotating machinery fault diagnosis, Mechanical Systems and Signal Processing 224 (2025) 111950.

[10] J. Yan, Y. Cheng, F. Zhang, M. Li, N. Zhou, B. Jin, H. Wang, H. Yang, W. Zhang, Research on multi-modal techniques for arc detection in railway systems with limited data, Structural Health Monitoring (2025) 14759217251336797.

[11] G.-Q. Zeng, J.-M. Shao, K.-D. Lu, G.-G. Geng, J. Weng, Automated federated learning-based adversarial attack and defence in industrial control systems, IET Cyber-Systems and Robotics 6 (2) (2024) e12117.

[12] H.-N. Wei, G.-Q. Zeng, K.-D. Lu, G.-G. Geng, J. Weng, Moar-cnn: Multi-objective adversarially robust convolutional neural network for sar image classification, IEEE Transactions on Emerging Topics in Computational Intelligence (2024).

[13] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Vol. 9, Foundations and Trends in Theoretical Computer Science, 2014.

[14] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR, 2017, pp. 1273–1282.

[15] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recognition 84 (2018) 317–331.

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

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2018).

[18] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, L. Daniel, Evaluating the robustness of neural networks: An extreme value theory approach, in: International Conference on Learning Representations (ICLR), 2018.

[19] S. Gowal, K. Dvijotham, R. Stanforth, T. A. Mann, P. Kohli, On the effectiveness of interval bound propagation for training verifiably robust models, in: NeurIPS, 2018, pp. 4899–4908.

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

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

[22] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[23] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. rep., University of Toronto (2009).

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

[25] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100,000+ questions for machine comprehension of text, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016, pp. 2383–2392.

[26] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, Glue: A multi-task benchmark and analysis platform for natural language understanding, in: International Conference on Learning Representations (ICLR), 2018.

[27] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, A detailed analysis of the kdd cup 99 data set, Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (2009).

[28] N. Moustafa, J. Slay, Unsw-nb15: A comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp. 1–6.

[29] H. S. Anderson, P. Roth, Ember: An open dataset for training static pe malware machine learning models, arXiv preprint arXiv:1804.04637 (2018).

[30] R. Ronen, M. Radu, M. Feuerstein, E. Yom-Tov, M. Ahmadi, Microsoft malware classification challenge, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 785–794.

[31] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, in: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, 2008.

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

[33] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, M. Hein, Robustbench: a standardized adversarial robustness benchmark, in: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.
URL `https://openreview.net/forum?id=SSKZPJCt7B`

[34] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, F. Roli, Feature space adversarial attacks with limited information, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 467–482.

[35] A. Huang, S. Song, X. Li, S. Zhao, S. Ermon, Metapoison: Practical general-purpose clean-label data poisoning, in: Advances in Neural Information Processing Systems (NeurIPS), 2020.

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

[37] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, T. Goldstein, Poison frogs! clean-label poisoning attacks on neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 6103–6113.

[38] T. Gu, B. Dolan-Gavitt, S. Garg, Badnets: Identifying vulnerabilities in the machine learning model supply chain, arXiv preprint arXiv:1708.06733 (2017).

[39] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, X. Zhang, Trojaning attack on neural networks, in: Network and Distributed System Security Symposium (NDSS), 2018.

[40] J. Zhang, Q. Z. Sheng, L. Zhang, A. Alhazmi, Z. Li, On poisoning attacks to graph-based recommender systems, in: IEEE International Conference on Data Engineering (ICDE), IEEE, 2019, pp. 1208–1219.

[41] J. Geiping, F. Bauermeister, H. Droge, M. Moeller, Witches' brew: Industrial scale data poisoning via gradient matching, in: International Conference on Learning Representations (ICLR), 2021.

[42] A. Turner, D. Tsipras, A. Madry, Label-consistent backdoor attacks, arXiv preprint arXiv:1912.02771 (2019).

[43] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 2020, pp. 2938–2948.

[44] M. Goldblum, L. Fowl, J. Terry, L. Huang, X. Zhao, T. Goldstein, Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (11) (2022) 8503–8524.

[45] Y. Yao, H. Guo, Y. Gao, Y. Yarom, S. Nepal, Latent backdoor attacks on deep neural networks, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2041–2055.

[46] H. Li, X. Xu, K. Ren, Invisible backdoor attacks against deep neural networks, arXiv preprint arXiv:2110.03735 (2021).

[47] H. Li, X. Wang, X. Xu, K. Ren, Nar-hunter: Detecting and understanding neural architecture backdoors in a black-box setting, in: Proceedings of the IEEE Symposium on Security and Privacy (S&P), 2022.

[48] 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 (AISec), ACM, 2017, pp. 15–26.

[49] W. Brendel, J. Rauber, M. Bethge, Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, in: International Conference on Learning Representations (ICLR), 2018.

[50] Y. Dong, F. Liao, T. Pang, H. Hu, J. Zhou, X. Xu, J. Z. Li, Boosting adversarial attacks with momentum, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 9185–9193.

[51] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Universal adversarial perturbations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1765–1773.

[52] Y. Liu, X. Chen, C. Liu, D. Song, Delving into transferable adversarial examples and black-box attacks, arXiv preprint arXiv:1611.02770 (2016).

[53] 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.

[54] Z. Yang, B. Zhang, W. Chang, Q. Shi, S. Ma, Neural network inversion in adversarial setting via background knowledge alignment, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2019, pp. 225–240.

[55] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: IEEE Symposium on Security and Privacy (S&P), IEEE, 2017, pp. 3–18.

[56] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, M. Backes, Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models, in: Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS), 2019.

[57] S. J. Oh, M. Augustin, B. Schiele, M. Fritz, Towards reverse-engineering black-box neural networks, in: International Conference on Learning Representations (ICLR), 2019.

[58] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, N. Papernot, High accuracy and high fidelity extraction of neural networks, in: Proceedings of the 29th USENIX Security Symposium, 2020, pp. 1345–1362.

[59] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, T. Ristenpart, Stealing machine learning models via prediction apis, in: 25th USENIX Security Symposium, 2016, pp. 601–618.

[60] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients, Advances in neural information processing systems 32 (2019).

[61] V. Chandrasekaran, M. Jagielski, Z. S. Zhang, N. Carlini, D. Song, N. Papernot, Exploring connections between model extraction and active learning, in: Proceedings of the 2020 IEEE Symposium on Security and Privacy (S&P), IEEE, 2020, pp. 1339–1356.

[62] T. Orekondy, B. Schiele, M. Fritz, Knockoff nets: Stealing functionality of black-box models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4954–4963.

[63] K. Krishna, G. S. Tomar, A. P. Parikh, N. Papernot, M. Iyyer, Thieves on sesame street! model extraction of bert-based apis, in: International Conference on Learning Representations.

[64] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale (2017). arXiv:1611.01236.
URL `https://arxiv.org/abs/1611.01236`

[65] M. S. Ayas, S. Ayas, S. M. Djouadi, Projected gradient descent adversarial attack and its defense on a fault diagnosis system, in: 2022 45th International Conference on Telecommunications and Signal Processing (TSP), 2022, pp. 36–39. doi:10.1109/TSP55681.2022.9851334.

[66] 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.

[67] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings (2015). arXiv:1511.07528.
URL `https://arxiv.org/abs/1511.07528`

[68] J. Su, D. V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Transactions on Evolutionary Computation 23 (5) (2019) 828–841.

[69] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients (2019). arXiv:1906.08935.
URL `https://arxiv.org/abs/1906.08935`

[70] C. Guo, M. Rana, M. Cisse, L. van der Maaten, Countering adversarial images using input transformations, in: International Conference on Learning Representations (ICLR), 2018.

[71] P. Samangouei, M. Kabkab, R. Chellappa, Defensegan: Protecting classifiers against adversarial attacks using generative models, in: International Conference on Learning Representations (ICLR), 2018.

[72] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, M. I. Jordan, Theoretically principled trade-off between robustness and accuracy, in: International conference on machine learning, PMLR, 2019, pp. 7472–7482.

[73] J. M. Cohen, E. Rosenfeld, J. Z. Kolter, Certified adversarial robustness via randomized smoothing, in: International Conference on Machine Learning, PMLR, 2019, pp. 1310–1320.

[74] E. Wong, J. Z. Kolter, Scaling provable adversarial defenses, in: Advances in Neural Information Processing Systems (NeurIPS), 2018.

[75] R. Feinman, R. Curtin, S. Shintre, A. Gardner, Detecting adversarial samples from artifacts, in: IEEE European Symposium on Security and Privacy Workshops, 2017, pp. 36–42.

[76] W. Xu, D. Evans, Y. Qi, Feature squeezing: Detecting adversarial examples in deep neural networks, in: Network and Distributed System Security Symposium (NDSS), 2018.

[77] N. Liu, H. Yang, X. Hu, Adversarial detection with model interpretation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.

[78] S. Gupta, K. Ganju, X. Wang, S. Wang, Z. T. Kalbarczyk, R. K. Iyer, Model repair via neural reprogramming, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 1133–1150.

[79] Y. Wang, X. Ma, J. Bailey, F. Liu, S. Ji, M. E. Houle, J. Bailey, Meta learning for adversarial robustness, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[80] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems 32 (1) (2021) 4–24. doi:10.1109/TNNLS.2020.2978386.

[81] F. Tramer, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses, in: International Conference on Learning Representations, 2018.

[82] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 308–318.

[83] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017.

[84] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, H. Eichner, et al., Advances and open problems in federated learning, Foundations and Trends in Machine Learning 14 (1–2) (2021) 1–210.

[85] S. Hardy, W. H. Chen, J. Abo, Q. Chen, N. Cardozo, Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption, in: arXiv preprint arXiv:1711.10677, 2017.

[86] R. Rade, S.-M. Moosavi-Dezfooli, Reducing excessive margin to achieve a better accuracy vs. robustness tradeoff, in: International Conference on Learning Representations, 2022.

[87] J. Dong, S.-M. Moosavi-Dezfooli, J. Lai, X. Xie, The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24678–24687.

[88] J. Dong, L. Yang, Y. Wang, X. Xie, J. Lai, Toward intrinsic adversarial robustness through probabilistic training, IEEE Transactions on Image Processing 32 (2023) 3862–3872.

[89] M. Goldblum, L. Fowl, T. Goldstein, Adversarially robust few-shot learning: A meta-learning approach, Advances in Neural Information Processing Systems 33 (2020) 17886–17895.

[90] J. Dong, Y. Wang, J.-H. Lai, X. Xie, Improving adversarially robust few-shot image classification with generalizable representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9025–9034.

[91] J. Dong, Y. Wang, X. Xie, J. Lai, Y.-S. Ong, Generalizable and discriminative representations for adversarially robust few-shot learning, IEEE Transactions on Neural Networks and Learning Systems 36 (3) (2024) 5480–5493.

[92] J. Dong, P. Koniusz, J. Chen, X. Xie, Y.-S. Ong, Adversarially robust few-shot learning via parameter co-distillation of similarity and class concept learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 28535–28544.

[93] J. Dong, P. Koniusz, J. Chen, Z. J. Wang, Y.-S. Ong, Robust distillation via untargeted and targeted intermediate adversarial samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 28432–28442.

[94] C. Schlarmann, N. D. Singh, F. Croce, M. Hein, Robust clip: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models, arXiv preprint arXiv:2402.12336 (2024).

[95] J. Dong, P. Koniusz, X. Qu, Y.-S. Ong, Stabilizing modality gap & lowering gradient norms improve zero-shot adversarial robustness of vlms, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1, 2025, pp. 236–247.

[96] J. Dong, P. Koniusz, Y. Zhang, H. Zhu, W. Liu, X. Qu, Y.-S. Ong, Improving zero-shot adversarial robustness in vision-language models by closed-form alignment of adversarial path simplices, in: Forty-second International Conference on Machine Learning.

[97] J. Dong, Y. Wang, J. Lai, X. Xie, Restricted black-box adversarial attack against deepfake face swapping, IEEE Transactions on Information Forensics and Security 18 (2023) 2596–2608.

[98] J. Chen, J. Dong, X. Xie, Mind the trojan horse: Image prompt adapter enabling scalable and deceptive jailbreaking, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 23785–23794.

[99] J. Dong, J. Chen, X. Xie, J. Lai, H. Chen, Survey on adversarial attack and defense for medical image analysis: Methods and challenges, ACM Computing Surveys 57 (3) (2024) 1–38.

[100] J. Ebrahimi, A. Rao, D. Lowd, D. Dou, Hotflip: White-box adversarial examples for text classification, in: Annual Meeting of the Association for Computational Linguistics (ACL), 2018.

[101] N. Carlini, D. Wagner, Hidden voice commands, in: 25th USENIX Security Symposium (USENIX Security 16), USENIX Association, 2016, pp. 513–530.

[102] D. Jin, Z. Jin, J. T. Zhou, P. Szolovits, Is bert really robust? natural language attack on text classification and entailment, in: AAAI Conference on Artificial Intelligence, 2020.

[103] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, Z. M. Mao, Adversarial sensor attack on lidar-based perception in autonomous driving, in: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, 2019.

[104] Darpa assured autonomy program, https://www.darpa.mil/program/assured-autonomy.

[105] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, A. Y. Ng, Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network, Nature medicine (2019).

[106] S. Ghamizi, M. Cordy, M. Gubri, M. Papadakis, A. Boystov, Y. Le Traon, A. Goujon, Search-based adversarial testing and improvement of constrained credit scoring systems, in: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020.

[107] M. Barreno, B. Nelson, A. D. Joseph, J. D. Tygar, The security of machine learning, Machine learning (2010).

[108] M. Seydali, F. Khunjush, B. Akbari, J. Dogani, Cbs: A deep learning approach for encrypted traffic classification with mixed spatio-temporal and statistical features, IEEE Access 11 (2023) 141674–141702.

[109] P. Addesso, M. Cirillo, M. Di Mauro, V. Matta, Advoip: Adversarial detection of encrypted and concealed voip, IEEE Transactions on Information Forensics and Security 15 (2019) 943–958.

[110] H. Liu, J. Dani, H. Yu, W. Sun, B. Wang, Advtraffic: Obfuscating encrypted traffic with adversarial examples, in: 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS), IEEE, 2022, pp. 1–10.

[111] M. Zhan, J. Yang, D. Jia, G. Fu, Eapt: An encrypted traffic classification model via adversarial pre-trained transformers, Computer Networks 257 (2025) 110973.

[112] W. Hu, Y. Tan, Generating adversarial malware examples for black-box attacks based on gan, in: International Conference on Data Mining and Big Data, Springer, 2022.

[113] K. Grosse, N. Papernot, P. Manoharan, M. Backes, P. McDaniel, Adversarial examples for malware detection, in: European symposium on research in computer security, Springer, 2017.

[114] H. Hosseini, S. Kannan, B. Zhang, R. Poovendran, Deceiving google's perspective api built for detecting toxic comments, arXiv preprint arXiv:1702.08138 (2017).

[115] Z. Zhao, D. Dua, S. Singh, Generating natural adversarial examples, in: ICLR, 2018.

[116] Y. Liu, P. Ning, M. K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Transactions on Information and System Security (TISSEC) (2011).

[117] A. Keliris, H. Salehghaffari, B. Cairl, P. Krishnamurthy, M. Maniatakos, F. Khorrami, Machine learning-based defense against process-aware attacks on industrial control systems, in: 2016 IEEE International Test Conference (ITC), IEEE, 2016.

[118] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org, 2020.

[119] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in: International Conference on Machine Learning, PMLR, 2018, pp. 274–283.

[120] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami, Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, in: arXiv preprint arXiv:1605.07277, 2016.