© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711918

Adversarial Robustness and LLM Red Teaming: A Unified Review of Security Toolkits

S. MUTHUVEL[1], AKAASSH SUNDAR[2]
[1]*Assistant Professor, Dept. of CSE, AMET University, Chennai*
[2]*Undergraduate Student, Dept. of CSE (AI&DS), AMET University, Chennai*

Abstract- ***As advanced machine learning (ML) and large language model (LLM) systems are deployed at scale, the security perimeter has expanded to include both classical adversarial ML threats and LLM-specific risks such as prompt injection, jailbreaks, and sensitive information leakage. This paper presents a structured comparison of open-source and community toolkits spanning these domains, covering canonical robustness libraries and orchestration utilities for deployment alongside modern LLM and agent security tooling for attack automation, red teaming, and runtime defenses, plus adjacent capabilities in deception, reverse engineering, and data-centric audit/visualization.***

Index Terms- ***Adversarial robustness, AI security, red teaming, large language models (LLMs), jailbreaks, prompt injection, guardrails, Responsible AI governance, CI/CD integration***

I. INTRODUCTION

AI capabilities have moved from research labs into core business operations. That shift has elevated the importance of robustness against adversarial behavior—ranging from data and model manipulation in traditional ML to prompt-level attacks, jailbreaks, and leakage risks in LLM ecosystems. The practitioner's challenge is twofold: (i) understand what today's open-source is and community toolkits can actually do, and (ii) select combinations that fit enterprise constraints (governance, integration, cost, and scale). This review maps the current landscape and synthesizes lessons that help teams build testable and defensible AI systems. The tools are evaluated by focus area, attack/defense coverage, model targets, integration and extensibility, and operational fitness, highlighting strengths, trade-offs, and where each approach best fits. We find that practical programs combine classical ML robustness evaluation with LLM-focused offensive testing and policy-enforced guardrails integrated into CI/CD and SecOps. We also identify persistent gaps multilingual, multi-turn, and multimodal coverage nd call for shared benchmarks and community challenges to improve reproducibility and realism in enterprise settings.

II. METHODOLOGY

A survey was done on widely used adversarial and red-teaming frameworks along five dimensions:
2.1. Focus & Category (e.g., adversarial robustness vs. LLM/agent security)
2.2. Attack/Defense Coverage (evasion, poisoning, extraction, jailbreaks, guardrails, etc.)
2.3. Model Targets (traditional ML, generative LLMs, code/binaries)
2.4. Integrations & Extensibility (APIs, CI/CD, cloud/on-prem)
2.5. Operational Fitness (where they work well, where they struggle)

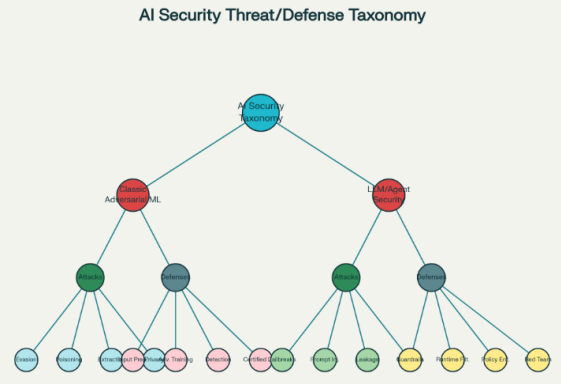

Figure 1:- AI Security Threat/Defense Taxonomy

III. COMPARATIVE EVALUATION

3.1. Classic Adversarial ML

3.1.1. IBM Adversarial Robustness Toolbox (ART).
ART is a Python ecosystem designed for evaluating and hardening non-generative ML systems across major frameworks (TensorFlow, PyTorch, Keras, scikit learn) [1, 2, 14]. It provides a broad catalog of attacks—e.g., gradient based methods such as FGSM/PGD and optimization driven approaches like Carlini–Wagner—alongside defenses, and

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711918

extends beyond evasion to include poisoning, extraction, and privacy oriented threats. ART's consistent APIs, multi modal coverage (image, audio, tabular), and enterprise friendly design make it a common anchor for ML robustness programs [1, 2, 14].

3.1.2. CleverHans.
CleverHans helped standardize how the community reports and reproduces adversarial ML results by offering reference implementations that became baseline comparators—especially in vision centric research [5, 6, 15]. While it generally exposes fewer integrations and less breadth than ART, its clarity and pedagogical value remain influential for experimentation and teaching.

3.1.3. Counterfit (Microsoft)
Rather than re implementing attacks, Counterfit provides a CLI orchestrator that drives assessments against live ML endpoints by leveraging underlying libraries (including ART and CleverHans) [7, 8]. This "glue" role makes it useful for quickly exercising deployed models across environments (cloud, edge, on prem), aligning well with red team workflows and repeatable testing.

3.2 LLM & Agent Security: Jailbreaks, Prompt Injection, and Guardrails

3.2.1. BrokenHill.
Bishop Fox's BrokenHill operationalizes automated jailbreak generation for LLMs using Greedy Coordinate Gradient (GCG) and related strategies to iteratively craft and evaluate adversarial prompts [3]. Its strength is throughput for offensive testing; its scope is intentionally narrow—focused on jailbreak payload engineering rather than comprehensive app security.

3.2.2. DreadnodecCrucible.
Crucible is positioned as an open red teaming/CTF platform that curates challenges, tracks outcomes, and supports both manual and automated tactics [16, 17]. Organizations use it for skills development, tool bake offs, and community benchmarking of attack techniques.

3.2.3. BurpGPT.
By bringing LLM analysis into traditional web security tooling, BurpGPT augments application assessments with generative reasoning about inputs, responses, and flow logic [4, 18, 19]. This bridges LLM risks (prompt injection, leakage) with existing AppSec processes—particularly relevant for AI powered web interfaces.

3.2.4. Guardrails.
Guardrails style frameworks provide policy driven validation and filtering around LLM I/O, including prompt injection indicators, content controls, and schema/contract enforcement to keep applications within defined bounds [11, 20]. They are designed to be embedded in production pipelines and tuned to domain specific rules.

3.2.5. Snyk (LLM/AgentSecurity).
Snyk Labs documents patterns for "red teaming your LLM agents" inside developer workflows, emphasizing CI/CD integration, automated tests for jailbreak susceptibility, and controls for supply chain style risks in agent tools and plug ins [12, 13]. This targets an increasingly important vector: LLM/agent behavior as code.

3.3. Deception, Reverse Engineering, and Data Visualization

3.3.1. Galah (LLM powered honeypot).
Galah generates dynamic, believable responses via LLMs to entice and observe attacker behavior on web endpoints, with optional integrations to network sensors like Suricata [9, 10]. It is valuable for discovering novel tactics, techniques, and procedures (TTPs), though sophisticated adversaries may fingerprint LLM generated artifacts if deployments are not carefully tuned.

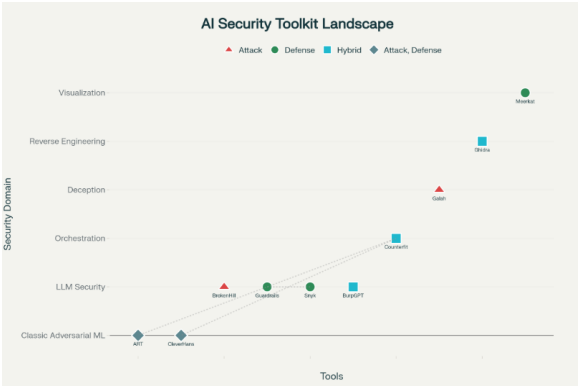

Figure 2:- AI Security Toolkit Landscape

3.3.2. Ghidra with LLM plugins (e.g., GhidrAssist).
Extensions for Ghidra apply LLMs to tasks such as summarizing functions, proposing deobfuscation

hints, or creating training corpora from reverse engineering workflows—bringing generative assistance to a classically manual discipline [21, 22]

3.3.3. Meerkat.

Meerkat emphasizes visualization first analysis using embeddings, clustering, and comparative views to audit datasets and detect fragility or distributional shifts that could amplify adversarial risk [23, 24]. It focuses on data diagnostics rather than attack generation, making it complementary to robustness toolkits.

IV. DISCUSSION AND SYNTHESIS

ART remains a comprehensive choice for robustness evaluation across modalities, while CleverHans continues to serve as a clean, research oriented baseline. Counterfit adds operational leverage by orchestrating attacks against deployed systems, facilitating realistic assessments that mirror production constraints [1, 2, 6, 7].

4.1. LLM Threat Landscape

The LLM stack has driven rapid specialization—automated jailbreakers (BrokenHill), red team challenge platforms (Crucible), and runtime defenses (Guardrails, Snyk) illustrate a shift toward toolchains that blend automated and human led tactics [3, 11–13, 16]. AppSec aligned utilities like BurpGPT fold LLM risks into established security lifecycles [4, 18].

4.2. Convergence with Traditional Security.

Honeypots (Galah), reverse engineering helpers for binaries (Ghidra plug ins), and data centric visualization (Meerkat) show how generative techniques are permeating adjacent security domains. The unifying trend is modular, API driven integration so these tools can fit CI/CD, SecOps, and governance controls without bespoke plumbing [9, 21, 22].

4.3. Gaps & Research Needs

Many LLM defenses are early stage; bypasses and transferability of attacks remain common, especially across languages, multi turn dialogues, and multimodal inputs [20, 25]. The field needs stronger shared benchmarks, reproducible datasets, and community exercises (e.g., CTFs) that reflect realistic threat models and deployment contexts [16, 17].

V. CONCLUSION

Modern AI security requires dual coverage: proven adversarial ML methods for traditional models and specialized LLM/agent defenses for generative systems. In practice, teams pair ART/CleverHans (analysis & baselines) with Counterfit (orchestration) to test classical ML, then extend with BrokenHill/Crucible (offense), Guardrails/Snyk (defense & SDLC), and BurpGPT (AppSec integration) for LLM applications. Deception (Galah), RE assistance (Ghidra plug ins), and data visualization (Meerkat) round out blue team visibility. Organizations should embed these tools behind governance gates, automate them in pipelines, and contribute findings to shared evaluations that raise the bar for robust and trustworthy AI.

REFERENCES

[1] Adversarial Robustness Toolbox (ART) – GitHub: https://github.com/Trusted-AI/adversarial-robustness-toolbox
[2] HiddenLayer – Unpacking the AI Adversarial Toolkit: https://hiddenlayer.com/innovation-hub/whats-in-the-box/
[3] Bishop Fox – BrokenHill (GCG jailbreak automation): https://bishopfox.com/blog/brokenhill-attack-tool-largelanguagemodels-llm
[4] Protecto – Best LLM Security Tools of 2025: https://www.protecto.ai/blog/best-llm-security-tools-safeguarding-large-language-models/
[5] CleverHans – GitHub: https://github.com/cleverhans-lab/cleverhans
[6] APXML – Adversarial ML Benchmarking Tools: https://apxml.com/courses/adversarial-machine-learning/chapter-6-evaluating-model-robustness/benchmarking-tools-frameworks
[7] Microsoft Security Blog – AI security risk assessment using Counterfit: https://www.microsoft.com/en-us/security/blog/2021/05/03/ai-security-risk-assessment-using-counterfit/
[8] ITPro – Microsoft launches Counterfit: https://www.itpro.com/technology/artificial-intelligence-ai/359409/microsoft-open-source-counterfit-to-stop-ai-hacks
[9] Galah (LLM Honeypot) – GitHub: https://github.com/0x4D31/galah

[10] Adel – Decoding Galah (YouTube): https://www.youtube.com/watch?v=XGsm4Qcc_Ag
[11] Confident AI – LLM Guardrails Guide: https://www.confident-ai.com/blog/llm-guardrails-the-ultimate-guide-to-safeguard-llm-systems
[12] Snyk Labs – Red Team Your LLM Agents: https://labs.snyk.io/resources/red-team-your-llm-agents-before-attackers-do/
[13] LinkedIn – Top 18 AI Red Teaming Tools: https://www.linkedin.com/posts/nidhal-shaikh_ai-technewsae-neweratech-activity-7363427060523945985-MxxG
[14] ART Docs – https://adversarial-robustness-toolbox.readthedocs.io
[15] CleverHans v2.1.0 – arXiv PDF: https://arxiv.org/pdf/1610.00768.pdf
[16] Dreadnode – Automation Advantage in AI Red Teaming: https://dreadnode.io/blog/the-automation-advantage-in-ai-red-teaming
[17] arXiv (2025) – Automation Advantage in AI Red Teaming: https://arxiv.org/html/2504.19855v1
[18] BurpGPT – Product site: https://burpgpt.app
[19] LinkedIn – BurpGPT Automated Vulnerability Detection: https://www.linkedin.com/pulse/burpgpt-chatgpt-powered-automated-vulnerability-detection-reddy
[20] Anthropic Docs – Mitigate jailbreaks and prompt injections: https://docs.anthropic.com/en/docs/test-and-evaluate/strengthen-guardrails/mitigate-jailbreaks
[21] GhidrAssist – GitHub: https://github.com/jtang613/GhidrAssist
[22] YouTube – Build an AI Powered Reverse Engineering Lab with Ghidra: https://www.youtube.com/watch?v=WOsVlzEXxJk
[23] Marktechpost (2025) – Top 18 AI Red Teaming Tools: https://www.marktechpost.com/2025/08/17/what-is-ai-red-teaming-top-18-ai-red-teaming-tools-2025/
[24] Stanford Hazy Research – Meerkat blog: https://hazyresearch.stanford.edu/blog/2023-03-01-meerkat
[25] arXiv (2025) – Bypassing LLM Guardrails: https://arxiv.org/abs/2504.11168I