From Research to Reality: Measuring the Adoption Lag of Adversarial Machine Learning Techniques in Industry Practice

Anonymous Authors
Paper #XXX
Unknown
redacted@institution.edu

Abstract

The adversarial machine learning (AML) research community has produced over a decade of publications on attacks and defenses, yet practitioners report persistent gaps between academic advances and industry deployment. While qualitative studies have documented this research-practice divide through surveys and interviews, no work has quantitatively measured the time lag from paper publication to demonstrable industry adoption. We address this gap using a novel *artifact-anchored backward traceability* methodology: starting from 9 authoritative industry artifacts—5 adversarial ML libraries (CleverHans, IBM ART, TextAttack, PyRIT, Foolbox), 3 standardized benchmarks (RobustBench, AutoAttack, HarmBench), and the MITRE ATLAS regulatory framework—we trace backward to extract and code 71 papers cited across these artifacts. Our approach provides verifiable adoption evidence through Git commit timestamps, benchmark integrations, and regulatory citations, enabling precise measurement of adoption lag (median: X.X years, IQR: X.X–X.X years). We find [key findings placeholder]. Our quantitative analysis reveals that [acceleration factors placeholder], with implications for research funding priorities, industry-academia collaboration models, and regulatory compliance timelines. This work provides the first systematic measurement of adversarial ML research-to-practice transfer, complementing prior qualitative gap studies with temporal adoption metrics.

CCS Concepts

• **Security and privacy → Malware and its mitigation**; • **Computing methodologies → Machine learning**.

Keywords

adversarial machine learning, technology adoption, research-to-practice gap, empirical measurement

ACM Reference Format:
Anonymous Authors. 2026. From Research to Reality: Measuring the Adoption Lag of Adversarial Machine Learning Techniques in Industry Practice. In *Proceedings of ACM Conference on Computer and Communications Security (CCS '26).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Adversarial machine learning has emerged as a critical security concern, with over a decade of academic research demonstrating vulnerabilities in ML systems [2, 8, 15, 26]. The field has produced hundreds of attack techniques, defense mechanisms, and robustness evaluations across computer vision [2], natural language processing [18], and more recently, large language models [16, 29]. Yet despite this substantial academic output, industry practitioners consistently report a persistent gap between research advances and operational deployment [1, 13, 17].

This research-practice divide manifests in concrete ways. Kumar et al.'s interviews with 28 organizations revealed that most ML engineers and incident responders are "not equipped with tactical and strategic tools to protect, detect and respond to attacks on their ML systems" [13]. Mink et al.'s qualitative analysis found that practitioners face barriers including "lack of institutional motivation and educational resources," "inability to adequately assess AML risk," and "organizational structures that discourage implementation" [17]. Most tellingly, Apruzzese et al. observed that while 89% of adversarial ML papers focus exclusively on deep learning and 63% evaluate only image data, "real-world evidence suggests that actual attackers use simple tactics to subvert ML-driven systems" [1].

These qualitative gap studies provide rich insights into *why* adoption barriers exist, but they cannot answer fundamental questions about *when* and *how fast* research translates into practice. How long does it take for a published attack technique to be implemented in industry tools? Do defenses get adopted faster than attacks? Has adoption accelerated over the field's evolution from foundational work (2014–2017) through expansion (2018–2021) to the LLM era (2022–2025)? What factors predict faster adoption—code availability, venue prestige, industry collaboration, or domain? Without quantitative temporal measurements, we cannot benchmark progress, identify bottlenecks, or design evidence-based interventions to accelerate research translation.

1.1 Our Approach: Artifact-Anchored Backward Traceability

We introduce a novel methodology to measure adversarial ML adoption lag through *reverse-engineering from authoritative industry artifacts.* Rather than starting from papers and speculating about impact through forward citation analysis, we begin with concrete evidence of industry adoption—widely-used tools, standardized benchmarks, and regulatory frameworks—and trace *backward* to the research papers they cite and implement.

Our approach selects 9 artifacts representing different adoption pathways: (1) **Tools**—CleverHans (6,401 GitHub stars), IBM Adversarial Robustness Toolbox (5,789 stars), TextAttack (3,348 stars), Microsoft PyRIT (3,343 stars), and Foolbox (2,936 stars); (2) **Benchmarks**—RobustBench (NeurIPS 2021, 750+ citations), AutoAttack (ICML 2020, 1,987 citations), and HarmBench (ICML 2024); (3) **Regulatory frameworks**—MITRE ATLAS (15 tactics, 66 techniques, 33 real-world case studies). These artifacts collectively represent the infrastructure through which adversarial ML research enters practice.

From these 9 artifacts, we automatically extracted 277 unique papers via Git repository scanning of all arXiv references, academic citations, and documentation. We then applied selection criteria prioritizing papers with strongest adoption evidence: 61 papers cited by 2+ artifacts (cross-validated adoption) and 10 papers cited only by MITRE ATLAS (regulatory adoption), yielding a final sample of **71 papers** for detailed coding. For each paper, we manually coded 12 variables capturing research characteristics (attack/defense/evaluation, threat type, domain, venue, code availability), threat model details (model access, gradient usage), and practical evaluation rigor.

Critically, our methodology provides *verifiable timestamps* for adoption events: Git commit dates when tools first reference papers, benchmark publication dates incorporating research, and MITRE ATLAS case study documentation dates. This enables precise calculation of adoption lag as the time difference between paper publication (conference date or first arXiv submission) and first adoption event across all artifacts. Our dataset provides ground truth for which papers achieved demonstrable industry adoption, when this occurred, and through which pathways.

1.2 Contributions

This work makes the following contributions:

- **First quantitative measurement** of adversarial ML research-to-practice adoption lag, spanning 2014–2025 across computer vision, NLP, LLMs, and regulatory domains.
- **Novel artifact-anchored methodology** using reverse citation tracing from 9 authoritative industry artifacts (5 tools, 3 benchmarks, 1 regulatory framework) to 71 research papers with verified adoption evidence.
- **Systematic coding framework** capturing 12 variables per paper (research type, domain, venue, code availability, threat model, practical evaluation), enabling analysis of factors predicting adoption speed.
- **Statistical analysis** using Kruskal-Wallis tests, Mann-Whitney U tests with Bonferroni correction, and Cox proportional hazards regression to identify adoption lag patterns by artifact type, publication era, and domain, plus acceleration factors including code availability, venue, and industry collaboration.
- **Reproducible dataset and analysis code** providing complete adoption timestamps, coding decisions with interrater reliability metrics, and statistical analysis scripts for community validation and extension.

1.3 Paper Organization

Section 2 provides background on adversarial ML threats and reviews qualitative gap studies. Section 3 formalizes our research questions. Section 4 details our artifact-anchored methodology, paper selection criteria, coding framework, and statistical analysis plan. Section 5 presents adoption lag measurements and domain comparisons. Section 6 interprets findings and discusses implications. Section 7 concludes with recommendations for accelerating research translation.

2 Background and Related Work

2.1 Adversarial Machine Learning Landscape

Adversarial machine learning encompasses threats across three primary attack categories, formalized in frameworks including MITRE ATLAS and NIST AI 100-2 [22].

Evasion attacks modify inputs at test time to cause misclassification while preserving semantic meaning. Foundational work includes Szegedy et al.'s discovery of adversarial examples [26], Goodfellow et al.'s Fast Gradient Sign Method (FGSM) [8], Carlini & Wagner's optimization-based attacks [2], and Madry et al.'s Projected Gradient Descent (PGD) [15]. Physical-world evasion attacks have demonstrated real-world risks including traffic sign misclassification [6] and autonomous vehicle manipulation.

Poisoning attacks corrupt training data or model parameters to degrade performance or insert backdoors. Key work includes backdoor attacks via training data manipulation [11] and federated learning poisoning. Recent concerns focus on supply chain vulnerabilities in foundation model training.

Privacy attacks extract sensitive information from trained models. Membership inference attacks [25] determine whether specific records were in training data, while model extraction attacks [27] reconstruct model parameters through black-box queries.

For large language models, new attack surfaces have emerged including prompt injection [9], jailbreaking [29], and alignment failures [28]. OWASP's LLM Top 10 ranks prompt injection as the #1 risk for LLM applications.

2.2 The Research-Practice Gap: Qualitative Evidence

Four seminal studies have documented the adversarial ML research-practice gap using qualitative methodologies, establishing the foundation that our quantitative work builds upon.

Kumar et al. (2020) [13] conducted interviews with 28 organizations across 11 industries, finding that practitioners "are not equipped with tactical and strategic tools to protect, detect and respond to attacks on their ML systems." Only 6 of 28 organizations were prepared to dedicate staff to building robust ML models. Their survey revealed concerning awareness gaps: most respondents lacked knowledge about even basic adversarial ML concepts.

Grosse et al. (2023) [10] provided the largest quantitative survey with 139 industrial practitioners, finding that approximately 5% of AI practitioners had experienced AI-specific attacks—remarkably low given academic attention. Their statistical analysis revealed that defense implementation correlated with threat exposure or expected likelihood of attack, not company size or organizational area. When asked about implementing countermeasures, the general consensus was "Why do so?"

Mink et al. (2023) [17] conducted 21 semi-structured interviews with data scientists and engineers, identifying three primary barriers: (1) lack of institutional motivation and educational resources for AML concepts, (2) inability to adequately assess AML risk, and (3) organizational structures discouraging implementation in favor of other objectives. Less than 25% of surveyed developers had access to security experts.

Apruzzese et al. (2023) [1] provided the most comprehensive analysis, examining 88 papers from top security venues (CCS, USENIX Security, NDSS, S&P) from 2019–2021 plus three real-world case studies. Their findings revealed stark research-practice misalignments: 89% of papers consider only deep learning, 63% evaluate only image data, only 5% address malware/phishing/intrusion detection, and 27% make no mention of computational costs. Their industry case studies demonstrated that real attackers use simple tactics (cropping, masking, stretching, blurring) rather than gradient-based optimization. They concluded: "Real-world evidence suggests that actual attackers use simple tactics to subvert ML-driven systems, and as a result security practitioners have not prioritized adversarial ML defenses."

2.3 Technology Adoption Measurement Methodologies

While no prior work has quantitatively measured adversarial ML adoption lag, established methodologies from other domains provide methodological foundations for our approach.

Citation lag analysis [20] measures knowledge diffusion speed through temporal analysis of citation patterns. Nakamura et al. introduced citation lag as the time difference between publication dates of cited and citing papers, demonstrating that inter-cluster citations have longer lags than intra-cluster citations, indicating different knowledge integration speeds across research areas.

Translational research lag studies provide critical benchmarks. Morris, Wooding & Grant's systematic review [19] synthesized 23 papers quantifying time lags in medical research, revealing substantial variation: publication to guideline (0–49 years, mean 8–17 years), drug discovery to commercialization (10–17 years, mean 12 years), and first description to highly cited (14–44 years, mean 24 years). They recommend using operational, measurable markers along translation pathways—directly analogous to our artifact-based adoption events.

Backward citation expansion [3] provides methodological justification for our reverse-engineering approach. Chen & Song's cascading citation expansion methodology enables "automatic expansion of an initial set by adding articles through citation links in forward, backward, or both directions," with backward expansion particularly useful when "a researcher may come across a recently published review article and would like to find previously published articles that lead to the state of knowledge summarized in the review"—precisely our use case with authoritative tools as "reviews" of implemented research.

Artifact-citation relationships have been validated for measuring research impact. Frachtenberg's analysis of 2,439 systems papers [7] found that papers with shared artifacts received 75% more citations than those without, with GitHub-hosted artifacts showing 86.7% availability vs. 77.8% for university-hosted. Heumüller et al.'s study of 789 ICSE papers [12] confirmed that "making artifacts publicly available has made a difference in terms of citations as a measure of scientific impact." These findings validate using artifact implementation as a proxy for research adoption.

2.4 Positioning This Work

Our work addresses a critical gap at the intersection of adversarial ML security and technology adoption measurement. While qualitative gap studies [1, 10, 13, 17] have documented *why* research doesn't translate into practice—organizational barriers, awareness deficits, threat model mismatches, computational costs—no work has measured *when* and *how fast* adoption occurs.

This temporal dimension is essential for evidence-based policy. Without quantitative lag measurements, we cannot benchmark whether adoption is improving, identify which research domains face the longest delays, or evaluate whether interventions (code release requirements, industry partnerships, standardized benchmarks) accelerate translation. Our artifact-anchored methodology provides verifiable adoption timestamps through Git commits, benchmark integrations, and regulatory citations, enabling the first systematic measurement of adversarial ML research-to-practice transfer timelines.

3 Research Questions

This work investigates three research questions examining adoption lag patterns, domain variation, and acceleration factors:

RQ1: Adoption Lag Measurement. What is the typical time lag between publication of landmark adversarial ML research and evidence of industry adoption, measured through tool integration, benchmark incorporation, and regulatory citation? We measure adoption lag in months from paper publication (conference date or first arXiv submission) to first adoption event (earliest Git commit, benchmark release, or MITRE ATLAS documentation). We stratify analysis by artifact type (tools, benchmarks, regulatory) and publication era (foundational 2014–2017, expansion 2018–2021, LLM 2022–2025) to identify temporal trends.

RQ2: Domain Variation. How does adoption speed vary across application domains (computer vision, natural language processing, large language models, malware detection, autonomous systems), and what factors explain these differences? We compare adoption lag distributions across 7 domains using pairwise Mann-Whitney U tests with Bonferroni correction ($\alpha = 0.05/21 = 0.0024$) and hypothesize that LLM research shows significantly shorter lags than computer vision due to heightened industry urgency around foundation model security.

RQ3: Acceleration Factors. What mechanisms—code availability, publication venue, industry collaboration, standardized benchmarks—predict faster adoption? We employ Cox proportional hazards regression modeling time-to-first-adoption with covariates including publication year (continuous), domain (categorical, reference: vision), venue type (ML vs. security, reference: ML conferences), code availability at publication (binary), and threat model assumptions (white/gray/black-box, reference: white-box). Hazard ratios >1 indicate faster adoption; we validate proportional hazards assumptions using Schoenfeld residuals.

4 Methodology

4.1 Artifact-Anchored Approach

Our methodology reverses the traditional research impact assessment approach. Rather than starting from papers and tracking forward citations to speculate about practical impact, we begin with concrete evidence of industry adoption—authoritative tools, benchmarks, and frameworks actively used by practitioners—and trace backward to the research papers they cite and implement.

This *artifact-anchored backward traceability* approach provides three key advantages: (1) **Verified adoption**—every paper in our sample has demonstrable evidence of industry use through tool implementation, benchmark integration, or regulatory citation; (2) **Precise timestamps**—Git commit dates, benchmark publication dates, and framework documentation provide verifiable adoption event timing; (3) **Multiple pathways**—we capture diverse adoption mechanisms including open-source tools, academic benchmarks, and regulatory frameworks, providing comprehensive coverage of research translation routes.

4.2 Artifact Selection

We selected 9 artifacts representing authoritative industry adoption pathways across three categories, applying rigorous inclusion criteria to ensure representativeness.

4.2.1 Open-Source Tools (5 artifacts). Tool selection criteria: (1) ≥1,000 GitHub stars indicating substantial community adoption; (2) Active maintenance with commits in 2024–2025; (3) Focus on adversarial ML specifically (not general ML security); (4) Multiple framework support or domain coverage.

- **CleverHans** (6,401 stars): First major AML library, created October 2016 by Ian Goodfellow (Google Brain/OpenAI) and Nicolas Papernot. Maintained by CleverHans Lab at University of Toronto. Provides reference implementations for foundational attacks (FGSM, PGD, C&W) across JAX, PyTorch, TensorFlow [23].
- **IBM Adversarial Robustness Toolbox** (5,789 stars): Enterprise-focused library created July 2018, donated to Linux Foundation AI & Data in 2020. Supports 9 ML frameworks, covers all threat types (evasion, poisoning, extraction, inference). Used in DARPA GARD program and DoD testing [21].
- **TextAttack** (3,348 stars): Dominant NLP adversarial framework, published EMNLP 2020 by QData Lab (UVA). Implements 16 attack recipes with HuggingFace integration. 835+ citations [18].
- **PyRIT** (3,343 stars): Microsoft's LLM red-teaming framework, released February 2024. Used for 100+ internal red teaming operations of generative AI models before public release. Integrates with Azure AI evaluation [14].
- **Foolbox** (2,936 stars): Academic benchmark tool from Bethge Lab (Tübingen), dual peer-reviewed publications (JOSS 2020, ICML 2017). Emphasizes minimum perturbation measurement and scientific rigor [24].

4.2.2 Standardized Benchmarks (3 artifacts). Benchmark selection criteria: (1) Peer-reviewed publication at top-tier ML venue (NeurIPS, ICML); (2) Community adoption evidenced by citations or leaderboard submissions; (3) Standardized evaluation protocols.

- **RobustBench** (NeurIPS 2021): Standardized adversarial robustness leaderboard with 750+ citations, 120+ evaluated models. Uses AutoAttack for consistent evaluation across CIFAR-10, CIFAR-100, ImageNet [4].
- **AutoAttack** (ICML 2020): Parameter-free attack ensemble (APGD-CE, APGD-DLR, FAB, Square Attack) with 1,987 citations. Revealed 13 of 50+ published defenses had robust accuracy overestimated by >10% [5].
- **HarmBench** (ICML 2024): First standardized LLM jailbreak evaluation framework. Covers 510 harmful behaviors, 18 attack methods, 33 target LLMs. Backed by Center for AI Safety [16].

4.2.3 Regulatory Framework (1 artifact).

- **MITRE ATLAS**: Industry-standard adversarial ML threat taxonomy (15 tactics, 66 techniques, 33 case studies). Co-created with Microsoft in 2020, now with 16 member organizations. $20M NIST partnership (December 2025). Explicitly referenced in EU AI Act alignment and CISA guidance.

4.3 Paper Extraction and Selection

We employed automated extraction followed by manual selection based on adoption evidence strength.

4.3.1 Automated Extraction (277 papers). For each of the 9 artifacts, we:

1. Cloned the complete Git repository history
2. Scanned all files (code, documentation, README, citations) for arXiv identifiers using regex pattern `arxiv.org/abs/\d+\.\d+`
3. Extracted academic citations from published benchmark papers (RobustBench, AutoAttack, HarmBench)
4. Retrieved MITRE ATLAS case study citations from framework documentation
5. Deduplicated across artifacts to create initial pool of 277 unique papers

4.3.2 Selection Criteria (71 papers). From the 277-paper pool, we applied two selection criteria prioritizing strongest adoption evidence:

Criterion 1: Multi-artifact papers (n = 61). Papers cited by ≥2 artifacts demonstrate cross-validated adoption across different industry pathways (e.g., implemented in both CleverHans and IBM ART, or cited by both RobustBench and MITRE ATLAS). This criterion ensures robust adoption signal.

Criterion 2: MITRE ATLAS-only papers (n = 10). Papers cited exclusively by MITRE ATLAS represent regulatory adoption pathway. While not implemented in tools or benchmarks, their inclusion in the industry-standard threat framework indicates practitioner awareness and relevance for compliance.

Final sample: **71 papers** with verified adoption evidence spanning 2014–2025.

4.4 Coding Framework

For each of the 71 papers, we manually coded 12 variables across three groups, following a structured codebook with explicit decision rules (see coding_instructions.pdf in reproducibility materials).

4.4.1 Research Characteristics (G1–G6).

- **G1 - Type**: Attack / Defense / Evaluation (primary contribution)
- **G2 - Threat**: Evasion / Poisoning / Privacy / N/A (attack category per NIST taxonomy)

- **G3 - Domain**: Vision / NLP / LLM / Malware / Audio / Tabular / Cross-domain (primary evaluation domain)
- **G4 - Venue**: ML conference / Security conference / Journal / arXiv-only (publication type)
- **G5 - Code available**: Yes / No (code link exists at time of coding)
- **G6 - Code timing**: At-publication / Post-publication / Never (when code released; "at-publication" = within 1 month of paper date)

4.4.2 Threat Model (T1–T2, Attack Papers Only).

- **T1 - Access level**: White-box / Gray-box / Black-box (model access assumptions; white = weights/gradients, gray = surrogate, black = queries only)
- **T2 - Gradient required**: Yes / No (whether gradients used at any attack stage)

4.4.3 Practical Evaluation (Q1).

- **Q1 - Real-world evaluation**: Yes / Partial / No ("Yes" = tested on production systems like Google API, Tesla, ChatGPT; "Partial" = realistic simulation; "No" = CIFAR/ImageNet only)

4.4.4 Coding Procedure. Two coders independently coded all 71 papers following the structured codebook. Initial coding was performed by GPT-4o with prompt-engineered instructions, then manually verified and corrected by human coders, resulting in 39 corrections documented in `coding_corrections.csv`. Inter-rater reliability was assessed using Cohen's κ across all 12 variables. Disagreements were resolved through discussion and consultation of paper full text.

4.5 Adoption Event Definitions and Lag Calculation

We define three types of adoption events with specific timestamp sources:

Tool adoption: Date of first Git commit that references the paper in code, documentation, or citations file. We extracted commit timestamps (UTC) using `git log -all -grep` for paper titles and arXiv IDs, verified through manual inspection.

Benchmark adoption: Publication date of the benchmark paper (conference proceedings date) that cites the research. For RobustBench (NeurIPS 2021), AutoAttack (ICML 2020), and HarmBench (ICML 2024), we use official conference dates.

Regulatory adoption: Date when MITRE ATLAS case study or technique description citing the paper was first published in framework documentation (extracted from GitHub repository history of `mitre/advmlthreatmatrix`).

For each paper, we record *all* adoption events across the 9 artifacts, then identify the **first adoption** as the earliest event across all pathways. Adoption lag is calculated as:

$$\text{Adoption Lag (months)} = \text{Date}_{\text{first adoption}} - \text{Date}_{\text{publication}} \quad (1)$$

Where $\text{Date}_{\text{publication}}$ is the earlier of: (1) conference/journal publication date, or (2) first arXiv submission date. All dates standardized to YYYY-MM-DD format, with lag calculated in months for consistency with translational research literature [19].

4.6 Statistical Analysis Plan

We employ non-parametric tests and survival analysis to address our research questions, with significance threshold $\alpha = 0.05$ (Bonferroni-corrected for multiple comparisons where applicable).

4.6.1 RQ1: Adoption Lag Measurement. **Descriptive statistics**: Median, interquartile range (IQR), range, and mean of adoption lags across full sample ($n = 71$).

Stratification by artifact type: Compare adoption lags for papers adopted through tools-only, benchmarks-only, regulatory-only, and multi-pathway adoption using Kruskal-Wallis test with post-hoc Dunn tests (Bonferroni-corrected).

Stratification by publication era: Compare adoption lags across three eras—foundational (2014–2017), expansion (2018–2021), LLM (2022–2025)—using Kruskal-Wallis test to identify temporal trends.

4.6.2 RQ2: Domain Variation. **Pairwise domain comparison**: Compare adoption lag distributions across 7 domains (Vision, NLP, LLM, Malware, Audio, Tabular, Cross-domain) using Mann-Whitney U tests with Bonferroni correction for 21 pairwise comparisons ($\alpha = 0.05/21 = 0.0024$).

Hypothesis test: LLM papers (n_{LLM}) show significantly shorter adoption lags than computer vision papers (n_{CV}) due to heightened industry urgency. One-tailed Mann-Whitney U test.

4.6.3 RQ3: Acceleration Factors. **Cox proportional hazards regression**: Model time-to-first-adoption using Cox regression:

$$\lambda(t|X) = \lambda_0(t) \cdot \exp(\beta_1 X_{\text{year}} + \beta_2 X_{\text{domain}} + \beta_3 X_{\text{venue}} + \beta_4 X_{\text{code}} + \beta_5 X_{\text{threat}}) \quad (2)$$

Where $\lambda(t|X)$ is the hazard rate (instantaneous adoption probability) at time t given covariates X:

- X_{year}: Publication year (continuous)
- X_{domain}: Domain (categorical: Vision, NLP, LLM, Malware, Audio, Tabular, Cross-domain; reference = Vision)
- X_{venue}: Venue type (ML conference vs. Security conference; reference = ML)
- X_{code}: Code available at publication (binary: Yes/No)
- X_{threat}: Threat model (White-box / Gray-box / Black-box; reference = White-box)

Hazard ratios $\exp(\beta_i)$ interpret as: HR > 1 indicates faster adoption, HR < 1 indicates slower adoption. We validate proportional hazards assumptions using Schoenfeld residuals and report 95% confidence intervals for all coefficients.

4.6.4 Software. All analyses conducted in Python 3.11 using: `pandas` (1.5.3) for data manipulation, `scipy` (1.10.1) for Kruskal-Wallis and Mann-Whitney U tests, `lifelines` (0.27.4) for Cox regression, `matplotlib` (3.7.1) and `seaborn` (0.12.2) for visualization. Analysis scripts and data available at [anonymized GitHub repository].

5 Results

5.1 Sample Characteristics

5.2 RQ1: Adoption Lag Patterns

5.3 RQ2: Domain Variation

5.4 RQ3: Acceleration Factors

6 Discussion

6.1 Key Findings Interpretation

6.2 Implications for Researchers

6.3 Implications for Practitioners

6.4 Limitations

6.5 Future Work

7 Conclusion

8 Acknowledgments

This work was supported by [funding agency redacted for review]. We thank [collaborators redacted for review] for valuable feedback on the methodology.

References

[1] Giovanni Apruzzese, Hyrum S. Anderson, Savino Dambra, David Freeman, Fabio Pierazzi, and Kevin A. Roundy. 2023. "Real Attackers Don't Compute Gradients": Bridging the Gap between Adversarial ML Research and Practice. In *Proceedings of the 1st IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 339–364.

[2] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 39–57.

[3] Chaomei Chen and Min Song. 2019. Visualizing a field of research: A methodology of systematic scientometric reviews. *PLOS ONE* 14, 10 (2019), e0223994. doi:10.1371/journal.pone.0223994

[4] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. 2021. RobustBench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[5] Francesco Croce and Matthias Hein. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 2206–2216.

[6] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1625–1634.

[7] Eitan Frachtenberg. 2022. Research artifacts and citations in computer systems papers. *PeerJ Computer Science* 8 (2022), e887.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations (ICLR)*.

[9] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec)*.

[10] Kathrin Grosse, Lukas Bieringer, Tarek R. Besold, Battista Biggio, and Katharina Krombholz. 2023. Machine Learning Security in Industry: A Quantitative Survey. *IEEE Transactions on Information Forensics and Security* 18 (2023), 1749–1762.

[11] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv preprint arXiv:1708.06733* (2017).

[12] Robert Heumüller, Sebastian Nielebock, Jacob Krüger, and Frank Ortmeier. 2020. Publish or perish, but do not forget your software artifacts. *Empirical Software Engineering* 25, 6 (2020), 4585–4616.

[13] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. 2020. Adversarial Machine Learning - Industry Perspectives. In *IEEE Security and Privacy Workshops (SPW)*. IEEE, 69–75.

[14] Gary D. Lopez Munoz, Amanda J. Minnich, Roman Lutz, Richard Lundeen, et al. 2024. PyRIT: A Framework for Security Risk Identification and Red Teaming in Generative AI Systems. *arXiv preprint arXiv:2410.02828* (2024).

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations (ICLR)*.

[16] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. In *Proceedings of the 41st International Conference on Machine Learning*, Vol. 235. PMLR, 35181–35224.

[17] Jaron Mink, Harjot Kaur, Juliane Schmüser, Sascha Fahl, and Yasemin Acar. 2023. "Security is not my field, I'm a stats guy": A Qualitative Root Cause Analysis of Barriers to Adversarial Machine Learning Defenses in Industry. In *32nd USENIX Security Symposium (USENIX Security 23)*. 3763–3780.

[18] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 119–126.

[19] Zoë Slote Morris, Steven Wooding, and Jonathan Grant. 2011. The answer is 17 years, what is the question: understanding time lags in translational research. *Journal of the Royal Society of Medicine* 104, 12 (2011), 510–520.

[20] Minoru Nakamura, Yuya Kajikawa, and Shintaro Suzuki. 2011. Citation lag analysis in supply chain research. *Scientometrics* 87, 2 (2011), 221–232.

[21] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. 2018. Adversarial Robustness Toolbox v1.0.0. *arXiv preprint arXiv:1807.01069* (2018).

[22] NIST. 2025. *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*. Technical Report NIST AI 100-2 E2025. National Institute of Standards and Technology.

[23] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Patrick McDaniel, et al. 2018. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv preprint arXiv:1610.00768* (2018).

[24] Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. 2020. Foolbox Native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX. *Journal of Open Source Software* 5, 53 (2020), 2607.

[25] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 3–18.

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing Properties of Neural Networks. In *2nd International Conference on Learning Representations (ICLR)*.

[27] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *25th USENIX Security Symposium*. 601–618.

[28] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How Does LLM Safety Training Fail?. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[29] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *arXiv preprint arXiv:2307.15043* (2023).