Towards Reliable Testing of Machine Unlearning

Anna Mazhar
Cornell University
Ithaca, NY, USA

Sainyam Galhotra
Cornell University
Ithaca, NY, USA

Abstract

Machine learning components are now central to AI-infused software systems, from recommendations and code assistants to clinical decision support. As regulations and governance frameworks increasingly require deleting sensitive data from deployed models, *machine unlearning* is emerging as a practical alternative to full retraining. However, unlearning introduces a software quality-assurance challenge: under realistic deployment constraints and imperfect oracles, how can we test that a model no longer relies on targeted information? This paper frames *unlearning testing* as a first-class software engineering problem. We argue that practical unlearning tests must provide (i) thorough coverage over proxy and mediated influence pathways, (ii) debuggable diagnostics that localize where leakage persists, (iii) cost-effective regression-style execution under query budgets, and (iv) black-box applicability for API-deployed models. We outline a causal, pathway-centric perspective, *causal fuzzing*, that generates budgeted interventions to estimate residual direct and indirect effects and produce actionable "leakage reports". Proof-of-concept results illustrate that standard attribution checks can miss residual influence due to proxy pathways, cancellation effects, and subgroup masking, motivating causal testing as a promising direction for unlearning testing.

1 Introduction

Machine learning models increasingly power features in *AI-infused software systems* including user-facing and backend capabilities such as search, recommendations, code assistants, and clinical triage. Engineering these systems differs from traditional software because model behavior depends on data, training pipelines, and deployment constraints that evolve over time [2]. Testing and quality assurance for ML components has consequently become an active topic in software engineering research, spanning test generation, adequacy criteria, and oracle challenges [3, 50].

These systems are often trained on proprietary and user-generated data that often contains sensitive information. Regulations such as the GDPR and CCPA now empower users to request the removal of their data [17, 41], forcing organizations to ensure that models can truly "forget" specified information. Beyond regulatory compliance, robust unlearning is vital for mitigating bias that undermines trust and safety, and it plays a key role in post-deployment debugging by exposing and correcting spurious correlations.

Retraining models from scratch after every deletion request is often impractical for large-scale systems. [7] This motivates the community to study approximate unlearning methods that update model parameters to (approximately) erase targeted information.

While these methods can be more efficient than full retraining, they introduce a fundamental software quality-assurance challenge: **under realistic deployment constraints, how can we test that unlearning has eliminated all relevant traces of the forgotten data?** This question is hard for the same reason many testing problems are hard: the desired property ("the model has forgotten Z") rarely has a perfect oracle, so practitioners must rely on systematic, indirect evidence [3].

Existing unlearning checks largely rely on data-sample removal or feature-level attribution. While useful, these approaches focus on surface associations and often fail to detect residual influence that persists indirectly through correlated proxies or mediated pathways, precisely the failure modes that matter in post-deployment testing.

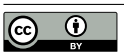
Example 1.1. Consider a lung disease diagnosis model trained on patient data including X-rays, patient metadata, and clinical notes. Suppose the model learns to associate `Hospital A` with higher disease risk because that hospital tends to treat more severe cases. A regulator subsequently mandates unlearning of hospital identifiers to prevent institutional bias. After unlearning, the model no longer reads visible hospital IDs or metadata.

Leakage through structured data. Even after removing hospital identifiers, the model may still infer institutional affiliation from correlated demographic attributes. For instance, features such as residential zip-code or employment history often indirectly encode where a patient is likely to receive care, As a result, the model continues to exhibit biased predictions, even though direct identifiers have been stripped away.

Leakage through unstructured data. A similar issue may arise with imaging data. Hospitals may configure their scanners differently, for example, using distinct grid settings, contrast levels, or reconstruction parameters. Even after unlearning hospital identifiers, the model may still recognize these scanner-specific signatures and associate them with higher disease risk.

Prior work on feature attribution and leakage analysis, including Shapley values and influence functions [20, 28, 30], can expose direct associations, but it does not verify whether residual influence persists through mediated or proxy pathways. These findings suggest that indirect signals, both structured or unstructured, can still encode residual influence, undermining the goal of unlearning.

This paper envisions a causal unlearning testing framework that moves beyond binary pass/fail judgments to form a feedback loop for improving unlearning methods. To be viable within routine software quality-assurance pipelines, such testing must satisfy four tightly coupled properties. **Thoroughness** requires detecting all practically relevant residual influence of the targeted information, including indirect effects that persist through correlated proxies or mediated causal pathways. **Debuggability** ensures that when residual influence is detected, tests explain where and how unlearning



FSE Companion '26, Montreal, QC, Canada

ACM ISBN 979-8-4007-2636-1/26/07
https://doi.org/10.1145/3803437.3805557

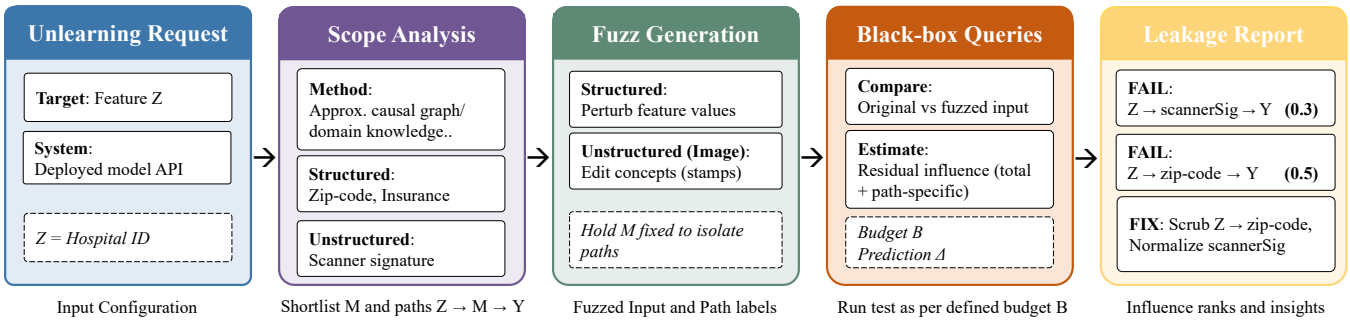


Figure 1: Causal fuzzing workflow: from unlearning request to actionable leakage report via path-guided input generation

failed to help localize the features or causal pathways responsible for leakage to enable targeted remediation. **Cost-effectiveness** makes this process sustainable in practice, allowing tests to run repeatedly (e.g., per deletion batch or model release) by prioritizing high-yield tests and reusing artifacts without incurring prohibitive computational or human cost. Finally, **black-box applicability** ensures deployability in real systems by relying only on input–output behavior, even when model internals are inaccessible, as in common API-based settings. These four properties follow directly from unlearning constraints in deployed ML systems.

To achieve these objectives, we propose framing unlearning testing as a *causal influence* problem: the claim "Z is forgotten" should mean that both its *direct* and *indirect* effects on predictions are negligible. Specifically, we propose a "causal fuzzing" mechanism to systematically test models and produce actionable, pathway-centric diagnostic insights under black-box access (Figure 1).

Our preliminary evidence on structured data shows that path-specific testing detects residual leakage that other methods miss, while remaining black-box. Looking ahead, we plan to extend the notion of mediator paths to natural language and image data using modality-specific perturbations. More broadly, we highlight a range of open research questions (§5) that call for community effort to advance rigorous, efficient, and generalizable unlearning testing.

2 Background and Related Work

SE has studied systematic testing of ML components using metamorphic relations and test generation, including DeepXplore [32], DeepTest [44], DeepHunter [46]. Recent work motivates black-box testing under deployment constraints, e.g., via test-case diversity [1], and advances metamorphic and contrastive testing for modern ML services [10, 26, 31, 45]. These techniques motivate our emphasis on efficiency and black-box applicability, but they do not target *forgetting*.

Unlearning Evaluation and Verification. Machine unlearning aims to update a model as if certain training data had never been used [7, 24]. A growing line of work studies how to evaluate/verify unlearning, including memorization-based metrics [25], stronger adversarial protocols [21, 40], and evidence that verification can be fragile or introduce new risks (e.g., dishonest providers and reconstruction attacks) [6]. Benchmarks and surveys further systematize evaluation (e.g., MUSE) [36, 50]. However, these evaluations typically provide aggregate evidence and lack pathway-level, debuggable tests under black-box constraints.

Influence and Leakage Analyses. A large body of work estimates data/feature influence via Shapley values, influence functions, and gradient/trajectory analyses [20, 28, 30], and uses attribution methods such as saliency maps and integrated gradients for unstructured models [39, 42]. Complementary privacy and extraction work shows that models can retain and reveal training information [8, 9, 11, 33, 34, 38]. Overall, these approaches often quantify surface exposure or direct association and may miss mediated leakage through proxies.

3 Desired Properties of Unlearning Tests

Machine unlearning introduces a *post-deployment* quality-assurance problem for AI-infused software systems: after an unlearning request, practitioners need *testable* evidence that the model no longer relies on the targeted information. In deployed settings this evidence must be collected under limited observability (e.g., API-only access) and imperfect oracles [2, 3]. From a testing perspective, unlearning checks are useful only if they (i) achieve *thoroughness* (high coverage) to avoid false reassurance when influence persists through proxies, (ii) are *debuggable* so failures localize *which* routes remain influential and guide remediation, (iii) are *cost-effective* enough to run routinely (e.g., per deletion batch or release), and (iv) remain *black-box applicable* because many models are accessible only via service interfaces. These properties represent a deployment-driven minimal set of necessary conditions for unlearning tests to function as first-class software tests in deployed settings.

3.1 Thoroughness (Coverage)

Unlearning tests should detect *all practically relevant* residual influences. For ML components, relying on a single simplistic coverage target is risky. Prior work shows that common DNN coverage criteria may correlate weakly with fault detection or model quality [22, 47]. Analogously, an unlearning test suite that checks only whether the *target feature* remains explicitly accessible can miss proxy-mediated influence.

Consider the testing implications for our hospital unlearning scenario (exp 1.1). A surface-level test might verify that hospital identifiers are no longer directly readable from model inputs or outputs. However, this is incomplete: the model may still route hospital-specific bias through demographic proxies (service-area indicators) or subtle imaging artifacts.

To ensure high coverage, testing must therefore probe these indirect channels systematically. For demographic features (structured), this involves testing whether zip codes, insurance types, or

referral patterns still enable hospital inference. For imaging data (unstructured), verification requires checking if scanner-specific noise patterns continue to influence predictions in hospital-correlated ways. In practice, this is also an *oracle* challenge: we rarely have a perfect "forgotten/not-forgotten" oracle in this domain, motivating tests that leverage structured invariants (e.g., metamorphic-style relations) rather than relying on labels alone [3, 12].

Effective unlearning verification therefore cannot rely on feature-by-feature checks. It requires a *pathway-centric* approach that traces how target information flows through the model and detects both direct dependencies and complex mediated effects.

3.2 Debuggability

Unlearning testing must detect both residual influence and guide mitigation. Debuggability requires actionable localization of *where* leakage persists, analogous to fault localization and delta debugging [27, 48, 49]. In our hospital example, removing explicit identifiers may not eliminate institution-specific bias. A non-debuggable test yields only a binary verdict (e.g., "hospital bias detected"), leaving developers uncertain whether leakage arises from demographics, imaging artifacts, or other proxies.

A debuggable test should report *which pathways* remain influential (e.g., leakage through scanner-specific grid patterns in image corners, with secondary leakage through geographic proxies). Such localization enables targeted interventions, such as preprocessing to normalize scanner signatures, or applying unlearning focused on specific mediators.

3.3 Cost-effectiveness

For unlearning verification to be practical, particularly in large-scale or frequently updated models, it must be cost-effective. Comprehensive tests may require thousands of probes spanning different modalities, languages, and inference patterns, which can be prohibitively expensive even for verifying a single datapoint. In large language models, the combinatorial explosion of possible prompts and contexts can compound into substantial costs. Hence, balancing cost-effectiveness with thoroughness is crucial. This motivates prioritization mechanisms (which pathways to test first), reuse of test artifacts across versions, and guided exploration strategies akin to fuzzing's budgeted search for high-value tests [29].

3.4 Black-Box Applicability

AI-infused systems are often API-only, with limited internal access. Consequently, unlearning tests should support black-box setting using only inputs and outputs. This aligns with a key software testing principle: feasibility under production observability [2].

Black-box applicability also supports governance and audit contexts where independent validation may be required, and aligns with operational monitoring expectations in risk-management frameworks [15, 16, 23, 43]. At the same time, black-box constraints create explicit tradeoffs: they can reduce diagnosability and thoroughness unless the testing strategy leverages structure to compensate.

4 Causal Perspective on Unlearning

Machine unlearning introduces a post-deployment testing problem: after an unlearning request, practitioners need evidence that the model no longer relies on the targeted information Z, despite limited observability (often API-only). We propose a *causal testing* view: unlearning is successful only if the *direct and indirect* causal influence of Z on the model output Y is negligible.

Test Oracle. We represent the deployed predictor as part of a structural causal model (SCM), where nodes correspond to observed inputs/features, intermediate representations, and outputs. Unlearning verification reduces to testing whether interventions on Z can still change Y, either directly or through mediators. This yields a concrete oracle: if Z is forgotten, then intervening on Z and propagating its downstream causal consequences should not meaningfully change Y. For each intervention, model outputs on original and intervened inputs are compared via the absolute change in prediction score (or class probability), and expected residual influence is estimated by Monte Carlo averaging. Residual influence is thresholded to decide whether unlearning is complete. The oracle assumes access to causal structure as a directed acyclic graph and the ability to sample realistic intervention values, while functional dependencies remain unknown.

Causal fuzzing workflow (how it operates). Figure 1 summarizes the causal fuzzing workflow. Given a target Z, we first generate a *budgeted* test suite of interventions that probes candidate influence routes from Z to Y: (i) *Scope & candidates:* identify a small set of likely mediators/proxies M (from the causal graph, domain knowledge, or simple proxy screening); (ii) *Interventions:* generate fuzz inputs (feature perturbations for structured data; concept edits for images/text) by intervening on Z (and optionally blocking or fixing selected mediators M) to isolate total and path-specific effects. (iii) *Black-box queries:* query the model on the original vs. intervened inputs and estimate residual effects (total and/or path-specific) under a fixed query budget; (iv) *Leakage report:* rank failing tests by effect size and report *which mediator/path* explains the residual influence. The report should be structured as ranked entries e.g., $Z \rightarrow \texttt{BMI} \rightarrow Y$ with a high estimated effect and `BMI` flagged for inspection.

Approximate structure may come from domain knowledge, or causal discovery (e.g., LiNGAM [37]). In practice, coarse structure can target likely mediator routes and support refutation/robustness checks to detect some graph misspecifications [35]. We discuss reporting graded assurance under graph uncertainty in Section 5.

How this achieves the desired properties. Causal fuzzing satisfies the desired properties (§3) as follows. *High coverage* comes from targeting *paths* from Z to Y (direct and mediated) rather than enumerating features in isolation. *Debuggability* comes from attributing a failure to specific mediator sets or paths (e.g., $Z \rightarrow M \rightarrow Y$), producing actionable localization. *Cost-effectiveness* comes from budgeted testing: prioritizing a small number of high-risk mediators (e.g., by causal proximity, proxy strength, domain risk, or prior failures) instead of exhaustively perturbing all features and reusing test artifacts across model versions, akin to guided fuzzing under a fixed budget [29]. *Black-box applicability* holds because causal testing requires only model queries on interventions.

4.1 Proof-of-Concept: Structured Data

Goal and setup. To validate feasibility, we ran proof-of-concept experiments on predictors representative of decision and health informatics software, trained on structured datasets commonly used in SE fairness-testing literature [13]: *Adult Income* and *Drug Consumption* (real-world) [4, 18], and *Heart-Disease* (semi-synthetic).

We compared against baseline testing approaches: `permutation importance` [19], `SHAP` [30], and fairness metrics [5]. We chose targets Z that are (i) explicitly sensitive (`gender`, `age`) or (ii) plausibly causal for outcomes (`smoking`, `BMI`).

For Adult Income, we used a coarse causal graph specified based on domain knowledge [14]; for Drug Consumption, we initialized a graph with LiNGAM [37] and retained only domain-plausible edges for testing; for the semi-synthetic Heart-Disease dataset, we used the known SCM structure. We treat these graphs as coarse structural hypotheses used to target tests, not as guaranteed ground-truth causal structures. Our goal is not exact causal identification, but evaluating whether testing remains useful without perfect causal knowledge. We discuss reporting graded assurance under graph uncertainty in Section 5.

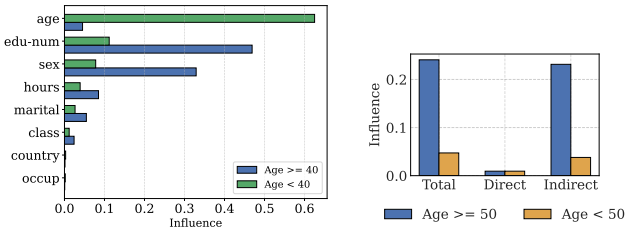
Unlearning baseline (to generate post-unlearning models). We use *feature removal* as a baseline unlearning operation (retrain without Z). This mirrors a common mitigation in practice and provides a clean setting to evaluate testing effectiveness: Z is no longer directly available, yet its influence may persist through correlated proxies or mediated routes. This simplified proxy for unlearning enables controlled evaluation of residual indirect effects, but it does not capture more realistic unlearning methods, which we leave for future work. Our focus is the *testing* strategy (causal fuzzing), rather than proposing a new unlearning algorithm.

4.1.1 Proxy pathways. A frequently observed failure mode is that unlearned information persists through alternative causal routes. After unlearning `smoking` in *Heart-Disease*, `SHAP` and `permutation importance` suggested that `smoking` influence was negligible. However, causal fuzzing revealed substantial residual influence mediated through `blood pressure` and `BMI`. This illustrates that removing Z does not prevent the model from re-expressing its influence via proxies even when surface-level attribution looks negligible.

4.1.2 Cancellation illusion. We observed that aggregate influence measures can underestimate residual dependence when opposing mediated effects cancel. In *Drug Consumption*, `Education` appeared weak overall in baseline evaluations. Path-specific analysis using causal fuzzing revealed why: two strong but opposing mediated pathways. Education simultaneously lowered risk by increasing health awareness and conscientiousness (*Cscore*), while increasing risk through elevated social exposure and extraversion (*Escore*). Although these effects nearly cancel in aggregate, the model retains both dependencies; small distributional shifts can disrupt the balance and re-activate residual influence. This blind spot motivates path-specific tests as a more faithful oracle for unlearning.

4.1.3 Subgroup masking. Global testing can obscure subgroup-specific vulnerabilities. In *Adult Income*, `Age` appeared only moderately influential overall, yet among individuals under 40 it became a dominant factor (Figure 2a). Likewise, in *Heart-Disease* after unlearning `BMI`, indirect influence was substantially stronger for older individuals and mediated through `blood pressure` (Figure 2b). In both cases, `SHAP` and `permutation importance` failed to capture this subgroup-specific influence variation.

Summary. These proof-of-concept results highlight that conventional verification can miss indirect, cancelling, and subgroup-specific effects, creating blind spots in unlearning assessment. By contrast, causal fuzzing yields pathway-level evidence that directly supports *coverage* and *debuggability*, while remaining compatible with black-box model access. As with other testing techniques, passing all unlearning tests does not prove absence of leakage, but systematically reduces known high-risk failure modes. Future work will scale the approach to higher-dimensional domains and study cost/coverage tradeoffs under explicit query budgets.



(a) Indirect effect of `Age` vs all features in Adult Income. **(b) Indirect effect of `BMI` on Heart Disease in older group.**

Figure 2: Mediated effects by age group.

5 Open Challenges and Future Directions

Our proof-of-concept illustrates that causal, pathway-centric tests can expose verification blind spots, but several challenges remain before such tests become routine regression artifacts in SE pipelines.

Imperfect causal knowledge. An open question is formalizing when such tests remain informative under graph uncertainty, i.e., how coverage and localization degrade as causal knowledge becomes partial. Promising directions include uncertainty-aware testing (e.g., testing families of plausible graphs) and reporting assurance levels rather than binary verdicts.

Budgeted regression testing. To support repeated unlearning in practice, future work should develop regression-style test suites that reuse perturbation artifacts across releases, prioritize high-risk pathways, and make cost/coverage tradeoffs explicit under budgets.

Unstructured and foundation models. Extending causal fuzzing beyond tabular data requires reliable modality-specific interventions (e.g., concept-level perturbations for images/text) and validity checks to ensure edits are realistic. In text setting, a key open problem is defining text-level mediators that remain meaningful under paraphrase and other surface changes. Another is designing realistic interventions that preserve task intent while perturbing candidate mediators, and deciding how residual influence should be estimated under a fixed prompt/query budget. For LLMs/VLMs, an additional challenge is prompt sensitivity; standardized prompt suites and pathway-oriented tests are needed to provide comparable evidence under black-box access.

6 Conclusion

In this paper, we have framed machine unlearning testing as a critical concern for for AI-infused software systems. We have identified key properties that practical unlearning tests must possess, including thoroughness, debuggability, efficiency, and black-box compatibility. Through our exploration of causal testing methods, we have demonstrated their potential to uncover residual influences and provide actionable insights for practitioners. Our vision is to establish rigorous testing frameworks that ensure genuine data removal while addressing the complexities of modern ML models.

References

[1] Zohreh Aghababaeyan, Manel Abdellatif, Lionel Briand, Ramesh S, and Mojtaba Bagherzadeh. 2023. Black-Box Testing of Deep Neural Networks through Test Case Diversity. *IEEE Trans. Softw. Eng.* (2023).

[2] Saleema Amershi et al. 2019. Software Engineering for Machine Learning: A Case Study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*.

[3] Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. 2015. The Oracle Problem in Software Testing: A Survey. *IEEE Transactions on Software Engineering* (2015).

[4] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository.

[5] Rachel K. E. Bellamy et al. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. arXiv:1810.01943

[6] Martin Andres Bertran, Shuai Tang, Michael Kearns, Jamie Heather Morgenstern, Aaron Roth, and Steven Wu. 2024. Reconstruction Attacks on Machine Unlearning: Simple Models are Vulnerable. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[7] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2020. Machine Unlearning. arXiv:1912.03817

[8] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*.

[9] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*.

[10] Jinyin Chen, Chengyu Jia, Yunjie Yan, Jie Ge, Haibin Zheng, and Yao Cheng. 2024. A Miss Is as Good as A Mile: Metamorphic Testing for Deep Learning Operators. *Proc. ACM Softw. Eng.* FSE (2024).

[11] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2021. When Machine Unlearning Jeopardizes Privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*. 896–911.

[12] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. 2018. Metamorphic Testing: A Review of Challenges and Opportunities. (2018).

[13] Zhenpeng Chen, Jie M. Zhang, Max Hort, Mark Harman, and Federica Sarro. 2024. Fairness Testing: A Comprehensive Survey and Analysis of Trends. *ACM Trans. Softw. Eng. Methodol.* (2024).

[14] Silvia Chiappa. 2019. Path-specific counterfactual fairness. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'19/IAAI'19/EAAI'19)*.

[15] European Commission (AI Act Service Desk). 2024. EU Artificial Intelligence Act — Article 11: Technical documentation. https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-11

[16] European Commission (AI Act Service Desk). 2024. EU Artificial Intelligence Act — Article 72: Post-market monitoring by providers. https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-72

[17] European Parliament and Council of the European Union. 2016. *General Data Protection Regulation (GDPR), Article 17.* Official Journal of the European Union. https://gdpr-info.eu/art-17-gdpr/

[18] Egan-Vincent Fehrman, Elaine and Evgeny Mirkes. 2015. Drug Consumption (Quantified). UCI Machine Learning Repository.

[19] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. 2019. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. arXiv:1801.01489

[20] Garima, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*.

[21] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. 2023. Towards Adversarial Evaluations for Inexact Machine Unlearning. arXiv:2201.06640

[22] Fabrice Harel-Canada, Lingxiao Wang, Muhammad Ali Gulzar, Quanquan Gu, and Miryung Kim. 2020. Is neuron coverage a meaningful measure for testing deep neural networks? *(ESEC/FSE 2020)*. New York, NY, USA.

[23] ISO/IEC. 2023. ISO/IEC 23894:2023 — Artificial intelligence — Guidance on risk management. https://www.iso.org/standard/77304.html

[24] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate Data Deletion from Machine Learning Models. arXiv:2002.10077

[25] Matthew Jagielski, Om Thakkar, Florian Tramèr, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Chiyuan Zhang. 2023. Measuring Forgetting of Memorized Training Examples. arXiv:2207.00099

[26] Weipeng Jiang, Juan Zhai, Shiqing Ma, Xiaoyu Zhang, and Chao Shen. 2024. COSTELLO: Contrastive Testing for Embedding-Based Large Language Model as a Service Embeddings. *Proceedings of the ACM on Software Engineering* (2024).

[27] J.A. Jones, M.J. Harrold, and J. Stasko. 2002. Visualization of test information to assist fault localization. In *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002.*

[28] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*.

[29] Caroline Lemieux and Koushik Sen. 2018. FairFuzz: A Targeted Mutation Strategy for Increasing Greybox Fuzz Testing Coverage. In *33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*.

[30] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 4768–4777.

[31] Yanzhou Mu, Juan Zhai, Chunrong Fang, Xiang Chen, Zhixiang Cao, Peiran Yang, Kexin Zhao, An Guo, and Zhenyu Chen. 2025. Improving Deep Learning Framework Testing with Model-Level Metamorphic Testing. *Proc. ACM Softw. Eng.* ISSTA (2025).

[32] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2019. DeepXplore: automated whitebox testing of deep learning systems. *Commun. ACM* (2019).

[33] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2020. Updates-leak: data set inference and reconstruction attacks in online learning. In *Proceedings of the 29th USENIX Conference on Security Symposium (SEC'20)*.

[34] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. arXiv:1806.01246

[35] Amit Sharma and Emre Kiciman. 2020. DoWhy: An End-to-End Library for Causal Inference.

[36] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. MUSE: Machine Unlearning Six-Way Evaluation for Language Models. In *The Thirteenth International Conference on Learning Representations*.

[37] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. 2006. A Linear Non-Gaussian Acyclic Model for Causal Discovery. *J. Mach. Learn. Res.* (2006).

[38] 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 (SP)*. 3–18.

[39] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034

[40] David M. Sommer, Liwei SOng, Sameer Wagh, and Prateek Mittal. 2022. Athena: Probabilistic Verification of Machine Unlearning. *Proceedings on Privacy Enhancing Technologies*.

[41] State of California. 2018. California Consumer Privacy Act of 2018 (CCPA). California Civil Code, Division 3, Part 4, Title 1.81.5. https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?sectionNum=1798.105&lawCode=CIV

[42] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*.

[43] Elham Tabassi. 2023. Artificial Intelligence Risk Management Framework (AI RMF 1.0). https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936225

[44] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. DeepTest: automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*. New York, NY, USA.

[45] Longtian Wang, Xiaofei Xie, Xiaoning Du, Meng Tian, Qing Guo, Zheng Yang, and Chao Shen. 2023. DistXplore: Distribution-Guided Testing for Evaluating and Enhancing Deep Learning Systems *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA.

[46] Xiaofei Xie et al. 2019. DeepHunter: a coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2019)*. Association for Computing Machinery, New York, NY, USA.

[47] Shenao Yan, Guanhong Tao, Xuwei Liu, Juan Zhai, Shiqing Ma, Lei Xu, and Xiangyu Zhang. 2020. Correlations between deep neural network model coverage criteria and model quality *(ESEC/FSE 2020)*. Association for Computing Machinery, New York, NY, USA.

[48] Andreas Zeller. 2002. Isolating cause-effect chains from computer programs *(SIGSOFT '02/FSE-10)*. Association for Computing Machinery, New York, NY, USA.

[49] A. Zeller and R. Hildebrandt. 2002. Simplifying and isolating failure-inducing input. *IEEE Transactions on Software Engineering* (2002).

[50] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. 2022. Machine Learning Testing: Survey, Landscapes and Horizons. *IEEE Transactions on Software Engineering* (2022).