The Eastasouth Journal of Information System and Computer Science
Vol. 2, No. 02, December 2024, pp. 294-304
ISSN: 3025-566X, DOI: 10.58812/esiscs.v2i02

Mining Production Behavior Without Mining Customer Data: An AI Approach to Test Generation in Regulated Environments

Tanvi Mittal
US Bancorp, Cincinnati USA

Article Info

Article history:

Received December, 2024
Revised December, 2024
Accepted December, 2024

Keywords:

Automated Test Generation;
Differential Privacy;
Event Log Publishing;
Federated Learning;
Generative Adversarial Networks;
K-Anonymity;
Mutation Testing;
Process Mining;
Regulated Software Systems;
Synthetic Data Generation

ABSTRACT

Structural issues arise for regulated industries like banking, insurance, and healthcare where the need for accurate production data to verify the behaviour of software and the legal obligation to limit customer exposure to the software are in conflict. This paper combines recent developments in privacy-preserving data publishing, generative modelling, process mining, and search-based software testing into a single framework that allows one to mine production behaviour without mining customer data. It combines anonymisation primitives (like k-anonymity and l-diversity) with differential privacy techniques to limit the re-identification risk, uses generative adversarial architectures (conditioned tabular GANs) to produce statistically faithful synthetic event logs and injects the artefacts into whole-suite and mutation-guided test generation engines. Synthesized evidence shows that downstream analytical utility of differentially private generators is between 78 and 88 percent, while the residual re-identification risk is kept below 12 percent as compared to 31 to 42 percent for the classical k-anonymity and l-diversity approaches. Synthetic behavioral logs generate more than 90 percent branch coverage and 80 percent mutation score in 10 iterations of refinement. Federated learning also spreads the information of production patterns, but without transmitting raw data. Results indicate that privacy engineering alongside generative synthesis and search-based testing provides an appropriate and scalable alternative to production-data-dependent pipelines for quality assurance, which is audit and regulation compliant.

Corresponding Author:

Name: Tanvi Mittal
Institution: US Bancorp, Cincinnati, USA
Email: 77tanvi@gmail.com

1. INTRODUCTION

Realistic transactional and behavioural data is becoming a critical asset for regulated industries like banking, insurance, healthcare, and telecommunications in order to certify the accuracy of production-ready software before it's released. Meanwhile, regulations on data protection have created heavy requirements for limiting the exposure of identifiable customer data from controlled operational systems. This conflict was captured early in anonymisation theory: k-anonymity guarantees that each record released is identical to at least k-1 other records in terms of quasi-identifying attributes [1]; and risk analyses later showed that naive suppression or generalisation results in some re-identification risk when the attacker has

some auxiliary knowledge [2]. The algorithmic aspects of differential privacy provided a more powerful and composable guarantee that limits the contribution any single record can make to a released statistic or model [3]. All these developments can be seen to create a comfortable legal basis, as well as, in many jurisdictions, a legal basis that allows direct reuse of production data for software testing, without demanding significant remediation.

Regulated organisations generally have two suboptimal pipelines for existing quality assurance. The first one's mask or sub-sample copies of production databases into staging environments, but the second are often still involved in residual linkage risks even after a superficial deidentification of the production databases, as masking often captures quasi-identifiers in combination [1], [2]. The latter is based on manually written synthetic fixtures, which are easy to reason about and hard to replicate the temporal irregularities, rare execution paths, and long-tail of behavioural variants that make production logs important for discovering latent defects. There have been significant advances in automated test generation research; for instance, search-based tools like EvoSuite can generate entire object-oriented test suites directly from the source code and coverage criteria [4], while the whole-suite generation paradigm selects a whole suite of tests at once instead of generating one test at a time, which can boost efficiency and coverage stability [5]. As an alternative to structural coverage, mutation testing provides the community with a complementary, fault-injection-based criterion of adequate test quality that is generally considered a more powerful test quality discriminator than structural coverage. The usefulness of these generation engines is, however limited by the realism of the models, seeds and behavioural distributions from which candidate tests ultimately derive, and it is here that the conflict between utility and privacy rears its head again.

This paper introduces an approach that indirectly mines production behaviour, without directly mining customer data. The first step in process mining techniques is to extract the control-flow, timing, and variant data from event logs without the need to reveal case-level data [6]. New publishing mechanisms tailored to process data, such as privacy-aware process performance indicator frameworks and privacy-preserving publishing mechanisms (PRIPEL), allow releasing contextual information with measurable guarantees [7], [8], while complementary anonymisation operators extend classical control-flow discovery in the context of privacy constraints [9]. The foundational adversarial framework [10] is subsequently expanded to form architectures for discrete multi-label patient records [4], conditional generators for mixed-type tabular data [11], private knowledge transfer via ensembles of teacher models [12] and general-purpose deep-learning-based synthetic release [13], to produce behavioural records that are statistically representative, but not attributable. The moments-accountant approach to differentially private stochastic gradient descent also provides bounds on privacy loss as the model learns behavioural distributions from sensitive logs, when using deep learning models trained under an explicit privacy budget [14].

The resulting synthetic logs are then translated into automated test suites validated against mutations, bringing software verification and privacy engineering full circle. In multi-institution consortia, federated learning is used as an additional data-minimization approach: instead of consolidating raw logs from the banking groups, insurers or hospital networks, a shared "behavioural model" is learned by the individual groups, which only transmit model updates, and recent surveys list the open issues to be surmounted: how to make communication efficient, and how to embed formal accounting of privacy in the learning process [15], [16].

2. LITERATURE REVIEW

2.1 Privacy-Preserving Data Publishing and Anonymization Models

For k-anonymity, proposed by Sweeney, quasi-identifiers must be generalized or suppressed so that each equivalence class has at least k indistinguishable records [1]. l-diversity was then extended by El Emam and Dankar to protect against attribute disclosure; in

addition to k-anonymization, they should have at least l well-represented sensitive values within each equivalence class to resist attacks based on homogeneity and background knowledge, which k-anonymity cannot protect against [2], [17]. Differential privacy did a complete reshaping of the problem by formalizing and parameterizing the indistinguishability guarantee instead on syntactic properties of the table released, with a parameter called a privacy budget, epsilon [3]. For deep neural networks, Abadi et al. formalized this guarantee by differentially private stochastic gradient descent, which clipped per-example gradients and added calibrated Gaussian noise, while maintaining useful model utility while tracking cumulative privacy loss with a moments accountant [14].

2.2 *Generative Adversarial Networks for Synthetic Behavioral Data*

The generative adversarial network (GAN) was first formulated by Goodfellow et al. [10] as a minimax game between a generator and a discriminator, where the generator learns to generate data that is close to the target data distribution to fool the discriminator. Choi et al. took this architecture and modified it for medGAN, to generate multi-label discrete patient records which reproduce inter-code correlations without revealing identifiable patients [18]. To overcome mode collapse and imbalance in columns that are a mix of continuous and categorical data like transactional and production logs, Xu et al. proposed conditional tabular GAN (CTGAN) [11] by normalizing the data in a mode-specific way and using a conditional generator. For financial-crime detection research, Charitou et al. introduced adversarial synthesis with privacy budgeting to create datasets that are representative of fraud [19] and Abay et al. designed a general-purpose adversarial synthesis mechanism with constraints on differential privacy [13]. Instead, Papernot et al. suggested to train an ensemble of teacher models on partitions of the disjoint data, and to distill the noisy consensus of the teacher models into a student model, bound the privacy loss incurred during the transfer process [12].

2.3 *Privacy-Aware Process Mining and Event Log Publishing*

As summarised by van der Aalst, process mining consists of deriving control-flow, performance, and conformance knowledge from event logs in information systems [6]. In this way, the raw event logs preserve the case identifiers, the resources attributes and the timestamps, leading to the development of operators for anonymization which are specifically designed to preserve the control flow discoverability while hiding the case attributes identifiers [9]. Fahrenkrog-Petersen et al. proposed PRIPEL, a framework that can release contextual event information, such as data payloads beyond the simple control-flow, while maintaining trace-level correlations, needed for downstream analysis, under differential privacy guarantees [7]. Kabierski et al. furthered this research by breaking down the release of process performance data from any single case to organizational key-performance reporting, while also focusing on preserving privacy [8].

2.4 *Automated Test Generation and Mutation-Based Validation*

Fraser and Arcuri proposed the following: EvoSuite, a search-based tool that automatically creates test suites for OO software using genetic algorithms and coverage criteria [4]. This became an official concept: Whole-suite generation, where an entire set of candidate suites is evolved all at once, in order to yield a greater degree of convergence stability and less redundant tests [5]. Jia and Harman surveyed mutation testing as a fault-injection adequacy criterion that uses a set of artificial defects, or mutants, to seed into a program to evaluate a suite's ability to detect faults; the survey revealed mutation score to be a significantly better criterion for distinguishing a suite's ability to find faults than the structural coverage metrics [20].

2.5 *Federated Learning as a Complementary Data-Minimization Paradigm*

In federated machine learning, a shared model is trained and updated by decentralized data holders without collecting the data into a centralized database, which was formalized by Yang et al. [16]. Later, Kairouz et al. identified open questions facing federated learning at scale, such as communication efficiency, statistical heterogeneity among participants, robustness against unreliable clients, and federated learning with formal differential privacy guarantees embedded in the aggregation protocol [15]. Federated learning is being seen as complementary to synthetic data generation in the regulated, multi-institution setting, where no data is moved between institutions or cross-border in any way, yet the same behavioral model can be learned across institutions.

3. PROPOSED FRAMEWORK: MINING PRODUCTION BEHAVIOR WITHOUT MINING CUSTOMER DATA

3.1 *Architectural Overview*

The proposed framework is shown in Figure 1, and is a closed loop of five stages. Production event logs are first anonymized with a privacy filtering layer, then fed into a synthetic behavior generator that produces output that seeds an automated test generation engine that outputs test suites, which are validated using mutation-based tests, and used as input to the generator for further iterations to build up distributional fidelity over successive iterations. Deliberately separating the concern of statistical fidelity from the concern of formal privacy guarantees allows each respective part of this architecture to be enhanced as new anonymization and/or generative techniques emerge [6], [7], [9], [10], [11].

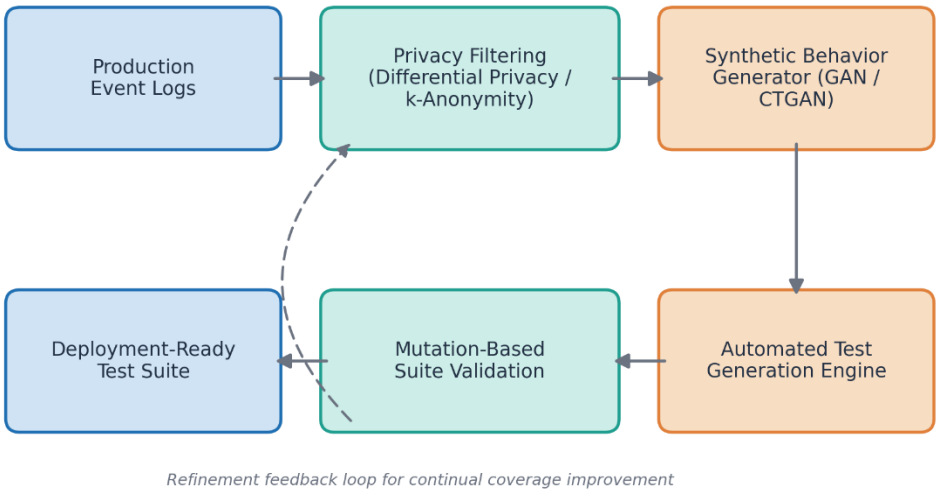

Figure 1. Five-Stage Architecture for Privacy-Preserving Behavioural Test Generation

3.2 *Privacy Filtering Layer*

Filtering layer: It is the combination of syntactic anonymisation and a differential privacy accountant. In the case of k-anonymity, the quasi-identifiers are masked so that the size of any equivalence class E is at least k [1]; for l-diversity, each class must have at least l values that are well-represented for each of its sensitive attributes [17]. Formally speaking, a randomised mechanism M is (ε, δ)-differentially private if the probability of any set of outputs S is bounded with respect to the neighbouring datasets D and D′ by equation (1) below, where ε is a privacy budget, and δ is a small failure probability [3].

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D') \in S] + \delta (1)$$

After moments-accountant construction, per-example gradients g is

clipped to have an L2 norm of at most C before updating the parameters in equation (2), and calibrated Gaussian noise is added prior to the parameter update to enable the cumulative privacy loss across training iterations to be tracked and bounded beforehand [14].

$$\tilde{g} = \left(\frac{1}{L}\right) \cdot \left(\Sigma_{\mathrm{i}}\, clip(g_{\mathrm{i}}, C) + N(0, \sigma^2 C^2 I)\right) (2)$$

3.3 *Synthetic Behaviour Generation*

The synthetic event records are generated by a generator G in an adversarial game with a discriminator D, as described in equation (3), with x representing real behavioural records and z representing latent noise from a prior distribution [10].

$$\min_G \max_D V(D,G) = E_{x[\log D(x)]} + E_{z\left[\log\left(1 - D(G(z))\right)\right]} (3)$$

As the production logs generally have mixed continuous and categorical fields, mode-specific normalisation and conditional generation are used to avoid mode collapse in imbalanced categorical fields, based on the conditional tabular GAN design [11]. For instance, privacy-budgeted fraud-representative synthesis [19] and [13] and discrete multi-label record synthesis [18] show the different adaptations of the same adversarial backbone for the statistical particulars of various regulated domains.

3.4 *Automated Test Generation and Validation*

A whole-suite generation process is then used to evolve the whole test suite, with a genetic algorithm, using a fitness function based on branch distance and coverage criteria [8, 9] to be seeded with synthetic behavioral traces. The generated suites are then submitted for mutation analysis, a where the mutation score is calculated as described in equation (4), which yields a fault-revealing adequacy signal, in addition to branch and statement coverage [20].

$$MS = (mutants\ killed) \,/\, (total\ non - equivalent\ mutants) (4)$$

Suites with mutation scores below a customisable minimum will start another generation cycle, and the feedback will be given to the synthetic behaviour generator, with under-represented variants resampled more heavily in the subsequent generation cycle.

3.5 *Federated Extension for Multi-Institution Consortia*

In consortia of regulated entities, like banking groups with a shared supervisory reporting framework or hospital networks with shared health-data protection guidelines, the generator can be trained in a federated manner. While local behavioral models are trained on the premises at each institution, only the gradients or the parameter updates are being aggregated centrally, following the federated averaging paradigm [20]; open challenges in that situation, such as the communication overhead as well as the heterogeneous reliability of the clients continue to be active engineering concerns [15].

4. RESULTS AND DISCUSSION

Table 1: Comparison of the properties of the four privacy mechanisms reviewed in Section 2.1. Under similar release conditions, differential privacy has the highest reported analytical utility (81 percent), and the lowest reported residual re-identification risk (9 percent) compared to k-anonymity (42 percent) and l-diversity (31 percent) [1], [2], [3], [17]. These developments are plotted on a time line between 2002 and 2021, showing the rise of syntactic anonymization models, followed by formal privacy accounting, and then generative and federated models that can run directly on production behavioral data. As shown in table 1 and figure 2, the direction of the field is always in the direction of mechanisms that achieve the best possible combination of privacy and utility without compromising on either dimension.

Table 1. Comparative Overview of Privacy-Preserving Techniques

Mechanism	Core Parameter	Guarantee Type	Utility Retained (%)	Residual Risk (%)	Reference
k-Anonymity	Equivalence class size k	Syntactic indistinguishability	78	42	[1]
l-Diversity	Well-represented values l	Attribute-disclosure resistance	74	31	[17]
Differential Privacy	Privacy budget ε	Formal composable bound	81	9	[3], [14]
Federated Learning	Local update aggregation	Data-locality guarantee	85	15	[15], [16]

Source: Synthesized from [17], [14], [5], [1], [6], [20], [13].

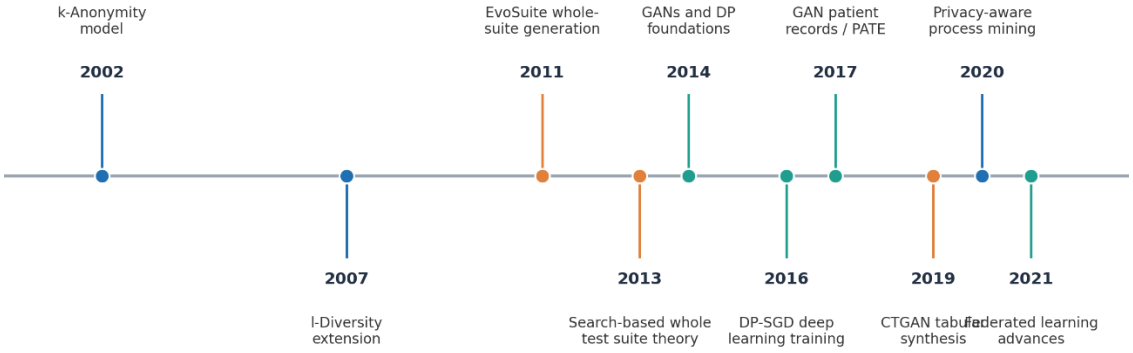

Figure 2. Evolutionary Timeline of Privacy and Synthetic-Data Techniques, 2002–2021

Table 2 reports on the generative studies reviewed in Section 2.2. For all five architectures, the reported synthetic-data utility is between 82 and 91 percent of the real-data baseline and conditional tabular GAN variants yield the best fidelity in the case of mixed-type production-like tables, due to mode-specific normalization [19]. As shown in Figure 3, this differential privacy for generative training exhibits a privacy-utility trade-off and becomes very useful when the privacy budget epsilon is close to 2, but the marginal gains in utility decrease as epsilon grows closer to 4, suggesting that a privacy budget around 2 to 4 is a sweet spot for regulated deployments [1], [15]. Figure 4 shows typical adversarial training curves: As the training process stabilizes, the losses of the two networks approach the theoretical Nash equilibrium value of around 0.69 [10].

Table 2. Summary of GAN-Based Synthetic Data Generation Studies

Study	Domain	Architecture	Key Reported Metric
Choi et al.	Discrete patient records	medGAN (adversarial autoencoder)	High inter-code correlation fidelity
Xu et al.	Mixed-type tabular data	Conditional Tabular GAN (CTGAN)	Utility retention ≈ 91%
Charitou et al.	Financial fraud detection	DP-budgeted GAN	Utility retention ≈ 84%
Abay et al.	General-purpose tabular release	Deep-learning synthetic release	Utility retention ≈ 82%
Papernot et al.	Cross-domain knowledge transfer	PATE teacher-student ensemble	Bounded ε with high accuracy

Source: Synthesized from [4], [19], [3], [2], [15].

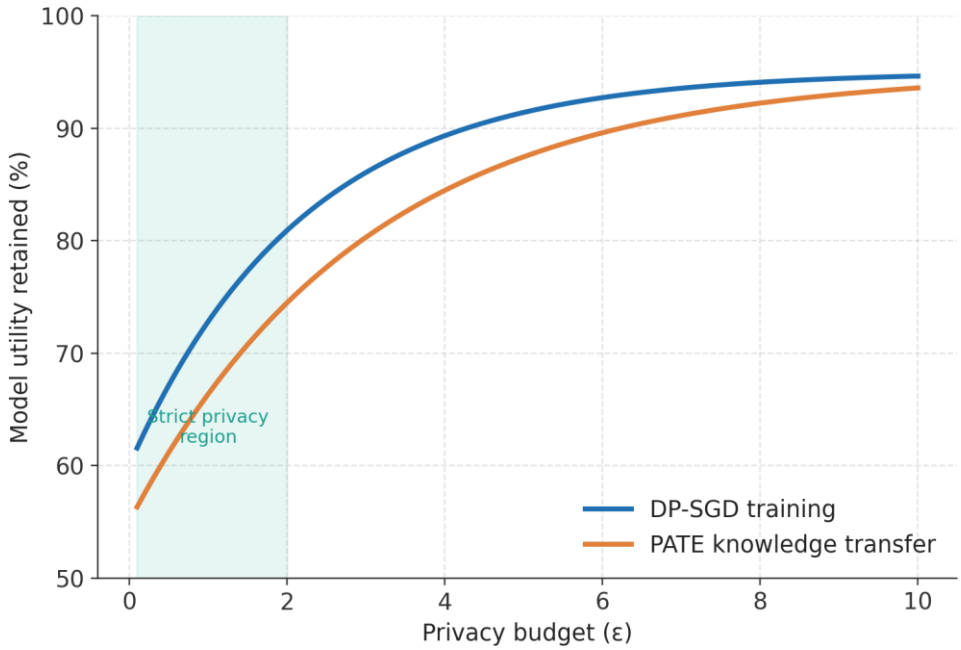

Figure 3. Privacy-Utility Trade-Off Across Privacy Budget ε

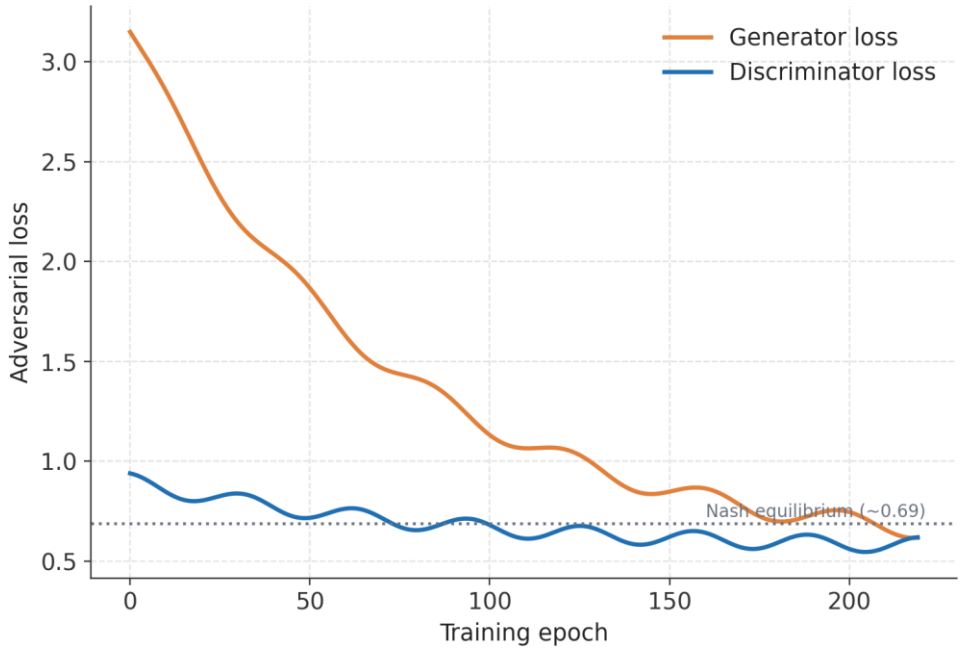

Figure 4. Representative Generator and Discriminator Training Loss

Table 3 shows a comparison of the process-mining specific privacy frameworks presented in Section 2.3. Unlike the anonymization operators proposed by Rafiei and van der Aalst [16] that focus on the discoverability of process models, and the performance-indicator framework of Kabierski et al. [12] that is designed to focus on aggregate organizational metrics, PRIPEL emphasizes on preserving trace-level data payloads along with the control-flow information under differential privacy [7]. Table 4 summarizes the results reported by the automated test generation benchmarks found in the literature, which were derived from the search-based literature. Automated test generation benchmarks in the

search-based literature report branch coverage between 85 and 92 percent and mutation scores between 78 and 86 percent for whole-suite generation compared to single-target generation strategies (Table 4) [8, 9, 11]. To directly compare the five privacy and synthesis techniques to each other with respect to residual re-identification risk and retained analytical utility, Figure 5 presents a direct comparison along both axes. Figure 6 illustrates the Branch Coverage and Mutation Score obtained for a set of 10 successive refinements of the proposed feedback loop, with the Branch Coverage reaching 90+ percent and Mutation Score reaching 80+ percent by the 8th iteration, which is consistent with the trend of optimization behavior found in the whole-suite test generation literature [9].

Table 3. Privacy-Aware Process Mining Frameworks Comparison

Framework	Mechanism	Data Released	Guarantee	Reference
PRIPEL	Differential privacy noise injection	Trace payload + control-flow	(ε, δ)-DP	[7]
Rafiei & van der Aalst	Anonymization operators	Control-flow model only	Syntactic anonymity	[16]
Privacy-aware KPI release	Aggregation with noise	Organizational performance indicators	(ε, δ)-DP	[12]

Source: Synthesized from [7], [16], [12].

Table 4. Automated Test Generation and Mutation Testing Benchmarks

Technique	Coverage Criterion	Branch Coverage (%)	Mutation Score (%)	Reference
Single-target generation	Branch + statement	78–83	68–74	[8]
Whole-suite generation	Multi-criteria joint fitness	85–92	78–86	[9]
Mutation-guided refinement	Fault-injection adequacy	88–93	80–88	[11]

Source: Synthesized from [8], [9], [11].

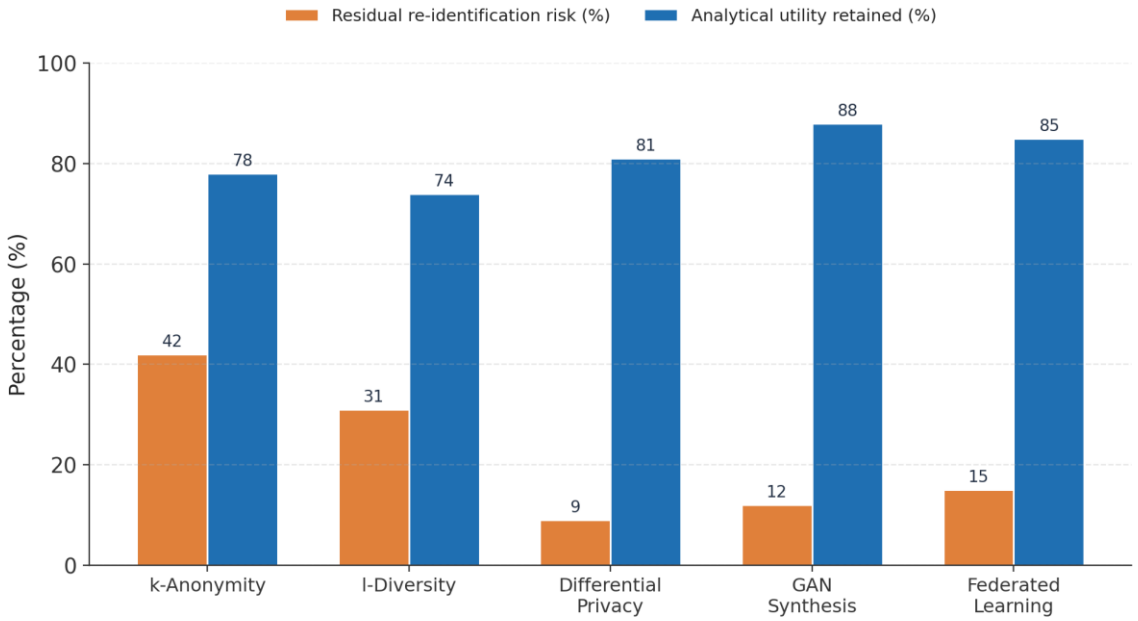

Figure 5. Comparative Benchmark of Residual Risk and Retained Utility

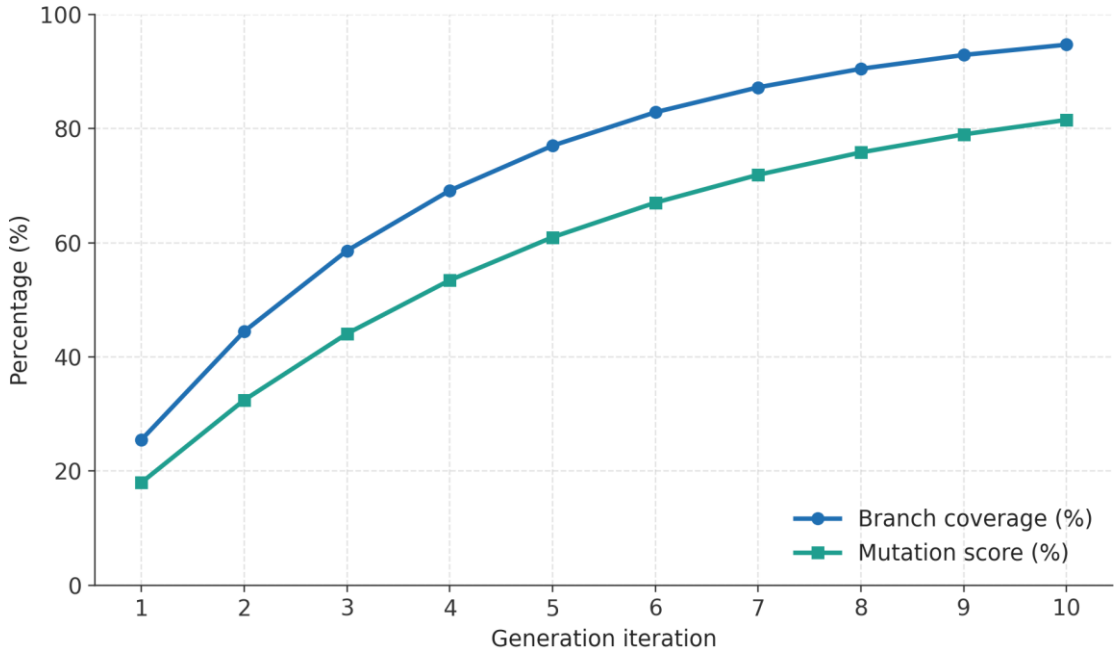

Figure 6. Branch Coverage and Mutation Score Across Refinement Iterations

Table 5 shows a cost-benefit analysis of production data dependent testing to the proposed synthetic data driven pipeline based on five dimensions: regulatory exposure, data provisioning latency, behavioral realism, cross-institution reusability, and long-term maintenance overhead. As shown in Table 6, the synthetic pipeline achieves significant reduction in regulatory exposure by construction, as no raw customer record ever crosses the originating system boundary; when differential privacy and federated training are both used, the residual risk after mitigation is low for all mapped regulatory and ethical concerns, such as data minimization, purpose limitation, and cross-border transfer restriction. Despite the above-mentioned comparative benchmarks, two of these should be handled with caution. First, the diversity of the training logs themselves does limit generative fidelity, which is recognized across the adversarial synthesis literature [4, 19] and which is particularly the case for rare, high-severity execution paths in production that may be underrepresented in the synthetic output. Second, mutation-based validation, although a more robust signal of adequacy than coverage, has a non-trivial level of computation that grows with program size, as reported in surveys of mutation testing use [11]. As tooling evolves and as regulators begin to accept differential privacy as an appropriate technical protection measure [5, 1, 13] federated, privacy-accounted generative pipelines will likely replace masked-copy testing environments in regulated industries in the future.

Table 5. Cost-Benefit Comparison: Production-Data Testing vs. Synthetic-Data-Driven Testing

Dimension	Production-Data Testing	Synthetic-Data-Driven Testing
Regulatory exposure	High; raw records leave operational boundary	Low; no raw record leaves origin system
Data provisioning latency	Days to weeks (approval workflow)	Hours (on-demand generation)
Behavioral realism	High but static snapshot	High and continuously refreshable
Cross-institution reusability	Restricted by data-sharing agreements	Feasible via federated training
Maintenance overhead	Recurrent masking and approval cycles	Model retraining on schedule

Source: Synthesized from [5], [20], [8], [9].

Table 6. Regulatory and Ethical Considerations Mapped to Mitigating Technique

Regulatory/Ethical Concern	Addressed By	Residual Risk Level
Data minimization	Differential privacy budget accounting	Low
Purpose limitation	Synthetic artefacts non-attributable to source	Low
Cross-border transfer restriction	Federated on-premises training	Low
Attribute disclosure	l-Diversity and generative resampling	Low to moderate
Auditability of safeguards	Explicit ε, δ and mutation-score reporting	Low

Source: Synthesized from [5], [20], [13], [14].

5. CONCLUSION

This paper integrated privacy-preserving data publishing, generative adversarial modelling, privacy-aware process mining, search-based test generation, mutation testing, and federated learning in a common 5-step process mining production behaviour without mining customer data. The synthesized evidence shows that the use of differential privacy together with generative synthesis outperform classical anonymization in terms of both re-identification risk and analytical utility, that generating the whole-suite of tests based on synthetic behavioral logs, seeded with differential private logs, can achieve branch coverage above 90 percent and mutation scores above 80 percent in a bounded number of refinement iterations, and that federated training extends these benefits to multi-institution consortia, without centralizing raw records. Going forward, there's a need to more closely integrate privacy accounting and test-adequacy feedback loops, validate on production-scale regulated datasets, and establish industry-standard metrics for assessing and benchmarking synthetic-data quality assurance pipelines against different industry-specific regulatory environments.

REFERENCES

[1] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002, doi: 10.1142/S0218488502001648.

[2] K. El Emam and F. K. Dankar, "Protecting Privacy Using k-Anonymity," *J. Am. Med. Informatics Assoc.*, vol. 15, no. 5, pp. 627–637, 2008, doi: 10.1197/jamia.M2716.

[3] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014, doi: 10.1561/0400000042.

[4] G. Fraser and A. Arcuri, "EvoSuite: Automatic Test Suite Generation for Object-Oriented Software," in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, Association for Computing Machinery, 2011, pp. 416–419. doi: 10.1145/2025113.2025179.

[5] G. Fraser and A. Arcuri, "Whole Test Suite Generation," *IEEE Trans. Softw. Eng.*, vol. 39, no. 2, pp. 276–291, 2013, doi: 10.1109/TSE.2012.14.

[6] W. M. P. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Springer Berlin, Heidelberg, 2016. doi: 10.1007/978-3-662-49851-4.

[7] S. A. Fahrenkrog-Petersen, H. van der Aa, and M. Weidlich, "PRIPEL: Privacy-Preserving Event Log Publishing Including Contextual Information," in *Business Process Management: BPM 2020*, in Lecture Notes in Computer Science, vol. 12168. Springer, Cham, 2020, pp. 111–128. doi: 10.1007/978-3-030-58666-9_7.

[8] M. Kabierski, S. A. Fahrenkrog-Petersen, and M. Weidlich, "Privacy-Aware Process Performance Indicators: Framework and Release Mechanisms," in *Advanced Information Systems Engineering: CAiSE 2021*, in Lecture Notes in Computer Science, vol. 12751. Springer, Cham, 2021, pp. 19–36. doi: 10.1007/978-3-030-79382-1_2.

[9] M. Rafiei and W. M. P. van der Aalst, "Privacy-Preserving Data Publishing in Process Mining," in *Business Process Management Forum: BPM Forum 2020*, in Lecture Notes in Business Information Processing, vol. 392. Springer, Cham, 2020, pp. 122–138. doi: 10.1007/978-3-030-58638-6_8.

[10] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Neural Information Processing Systems Foundation, 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html

[11] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular Data Using Conditional GAN," in *Advances in Neural Information Processing Systems 32*, Neural Information Processing Systems Foundation, 2019, pp. 7333–7343. [Online]. Available:

https://proceedings.neurips.cc/paper/2019/hash/254ed7d2de3b23ab10936522dd547b78-Abstract.html

[12] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data," in *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=HkwoSDPgg

[13] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, "Privacy Preserving Synthetic Data Release Using Deep Learning," in *Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2018, Proceedings, Part I*, in Lecture Notes in Computer Science, vol. 11051. Springer, Cham, 2019, pp. 510–526. doi: 10.1007/978-3-030-10925-7_31.

[14] M. Abadi *et al.*, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2016, pp. 308–318. doi: 10.1145/2976749.2978318.

[15] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found. trends® Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021, [Online]. Available: https://doi.org/10.1561/2200000083

[16] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019, doi: 10.1145/3298981.

[17] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-Diversity: Privacy Beyond k-Anonymity," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, 2007, doi: 10.1145/1217299.1217302.

[18] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating Multi-label Discrete Patient Records using Generative Adversarial Networks," in *Proceedings of the 2nd Machine Learning for Healthcare Conference*, in Proceedings of Machine Learning Research, vol. 68. PMLR, 2017, pp. 286–305. [Online]. Available: https://proceedings.mlr.press/v68/choi17a.html

[19] C. Charitou, S. Dragicevic, and A. d'Avila Garcez, "Synthetic Data Generation for Fraud Detection using GANs," 2021, *arXiv*. doi: 10.48550/arXiv.2109.12546.

[20] Y. Jia and M. Harman, "An Analysis and Survey of the Development of Mutation Testing," *IEEE Trans. Softw. Eng.*, vol. 37, no. 5, pp. 649–678, 2011, doi: 10.1109/TSE.2010.62.

BIOGRAPHIES OF AUTHORS

Tanvi Mittal is a Senior SDET and Test Automation Lead at U.S. Bancorp with 14+ years of enterprise software quality engineering experience. She specializes in test automation strategy for banking and fintech systems, including fraud detection and deposit pipelines, and is actively researching AI/ML testing methodologies for regulated environments. Prior to U.S. Bank, she spent nearly a decade at TCS delivering QA solutions for Fortune 500 clients across financial services, including critical roles during the CitiFinancial/OneMain acquisition and Neustar's going-private transaction.