Journal of Sustainability, Policy, and Practice

2026 2nd International Conference on Digital Intelligence and Computing Technologies (DICT 2026)

Article

Synthetic Tabular Data for Fraud Detection: A TSTR Comparison of Rule-Based, TVAE, and CTGAN Generators

Yunfei Gao[1,*]

[1] Electrical Engineering, Northwestern University, IL, USA
* Correspondence: Yunfei Gao, Electrical Engineering, Northwestern University, IL, USA

Abstract: Synthetic data offers a privacy-mitigating route to training anomaly detectors when labeled transactions are scarce, delayed, or barred from cross-institutional sharing. This study reports a controlled comparison of three tabular synthesizers, a rule-based simulator, a variational autoencoder (TVAE), and a conditional generative adversarial network (CTGAN), under the Train-on-Synthetic, Test-on-Real (TSTR) protocol. Each synthesizer is paired with three training strategies: standard supervision, weak supervision under simulated label delay, and semi-supervised fine-tuning with a small real sample, the last a controlled relaxation of strict TSTR. Detectors are evaluated on the European cardholder dataset released by the Université Libre de Bruxelles, with synthetic-to-real transferability assessed through a controlled distribution-shift sweep; IEEE-CIS Anomaly Detection and Bank Account Anomaly are used to characterize the cross-corpus statistical gaps that motivate the sweep rather than as separate retraining targets. CTGAN attains the closest statistical fidelity to real transactions, with a mean Kolmogorov--Smirnov distance of 0.092, and the strongest TSTR F1 of 0.68 when combined with semi-supervised fine-tuning, recovering roughly 82% of the real-data upper bound of 0.83. Weak supervision degrades more gracefully than standard supervision as the label-delay window widens from zero to fourteen days. Reliable transfer holds up to roughly a 10% synthetic-to-real statistical gap, beyond which all synthesizers lose F1, and the learned generators relinquish their advantage over the rule-based floor only as the gap approaches 40%. The results indicate that synthetic training is a workable but bounded substitute for real labels, with moderate rather than decisive gains, and that generator fidelity and a modest real anchor jointly govern transferability.

Keywords: financial anomaly detection; synthetic tabular data; train-on-synthetic-test-on-real; label delay

Received: 11 May 2026
Revised: 16 June 2026
Accepted: 30 June 2026
Published: 03 July 2026

1. Introduction

1.1. Background and Motivation

Payment fraud imposes a persistent and growing cost on the global financial system, with industry tracking placing worldwide card losses at roughly 33.8 billion U.S. dollars in 2023 and projecting continued expansion through the end of the decade [1]. The operational reality behind these figures is harsher than the aggregate suggests. Fraudulent transactions are rare, adversaries adapt quickly, and the confirmation of whether a flagged transaction was truly fraudulent often arrives days or weeks after the decision has to be made [2]. This verification latency, combined with extreme class imbalance and privacy regulations that restrict the pooling of transaction records across institutions, defines the working conditions under which any deployed detector must operate.

Recent work has begun to ask whether models can be trained when real labels are effectively unavailable, learning instead from synthetic transactions that imitate the

statistical signature of fraud without exposing protected customer data [3]. The appeal is direct. A bank constrained by the Gramm--Leach--Bliley Act or the California Consumer Privacy Act cannot freely share raw records, while it may be able to release or exchange a generative process that reproduces the relevant structure. Whether a detector trained purely on such data carries over to genuine transactions is an empirical question that has received limited systematic attention for tabular financial data, even as the underlying generators have matured.

1.2. Problem Statement and Research Questions

The Train-on-Synthetic, Test-on-Real (TSTR) protocol formalizes this question by fitting a detector entirely on generated data and reporting its performance on a held-out real sample. While TSTR has been examined for privacy-preserving release [4] and for sequential records, the comparative behavior of competing tabular synthesizers under the specific constraints of fraud detection, severe imbalance and delayed labels, remains underexplored. Standardized and reproducible evaluation pipelines for card fraud exist [5], yet they assume access to real labels and do not isolate the contribution of the data-generation step. This work does not propose a new architecture. It contributes a controlled comparative evaluation that holds the downstream classifier and the test data fixed while varying only the synthetic-data generator and the training strategy, allowing the transferability of each generator to be read directly.

1.3. Why TSTR Matters under Label Delay and Data Scarcity

Three questions organize the study. The first asks which family of generators, rule-based, autoencoder, or adversarial, produces synthetic transactions that transfer best to real data under a fixed detector. The second asks whether weak supervision that models delayed label arrival, or semi-supervised fine-tuning on a small real sample, can compensate for the absence of clean real labels. The third asks how large the statistical gap between synthetic and real data can grow before transferability collapses. These questions matter precisely because the conditions that make synthetic data attractive, scarcity and latency, are the same conditions under which a naive training procedure is most likely to fail quietly.

1.4. Scope and Contributions of This Comparative Evaluation

The evaluation makes three contributions. It assembles a side-by-side TSTR benchmark for tabular card fraud that spans rule-based, TVAE, and CTGAN generators under a shared protocol. It introduces an explicit label-delay simulation with windows ranging from 0 to 14 days, mapping how each training strategy degrades as confirmation lags. It evaluates synthetic-to-real transfer through a controlled distribution-shift sweep, drawing on three public datasets to characterize the statistical gaps that define the sweep's range, with every configuration documented to enable reproducible comparison. The framing is deliberately modest: the goal is to characterize the boundaries of synthetic training rather than to advance a single method.

2. Related Work

2.1. Synthetic Tabular Data Generation

Generating realistic tabular records is harder than it appears, because financial tables mix multimodal continuous columns with high-cardinality categorical fields and strong inter-column dependencies that simpler generative recipes tend to flatten.

2.1.1. Rule and Simulator-Based Generators

Simulator-driven approaches encode domain knowledge into an explicit generative process. The PaySim simulator reconstructs mobile-money transaction streams from aggregated real data using a multi-agent model, injecting fraud through scripted account takeover and transfer behavior [6]. Such generators are transparent and controllable, and they guarantee privacy by construction, while their coverage is bounded by the scenarios their designers anticipated, which limits the diversity of fraud patterns they can express and leaves them blind to drift in adversary behavior.

2.1.2. Deep Generative Models for Tabular Data

Learned generators instead estimate the data distribution directly. The conditional tabular GAN and its companion variational autoencoder address mode collapse on imbalanced categorical columns through conditional sampling and mode-specific normalization, and they remain the most widely adopted tabular baselines [7]. Both build on the variational autoencoder formulation that frames generation as latent-variable inference [8]. Diffusion-based synthesis has since been reported to have higher machine-learning utility than adversarial and autoencoder baselines across a range of public tables [9], although its cost and tuning burden are heavier. This study restricts its learned generators to TVAE and CTGAN, whose maturity and reproducibility make them appropriate reference points for a focused comparison rather than a survey of the synthesis frontier.

2.2. Fraud Detection under Imbalance and Label Delay

A large body of fraud-detection research models transactions as a graph and applies neural message passing to capture relational signals. Camouflage-aware detectors reweight neighbors to resist fraudsters who imitate legitimate behavior [10], and imbalance-aware sampling selects informative neighborhoods to counter skewed label distributions [11]. These graph methods report strong results, while they presuppose abundant labeled edges and a stable relational structure, assumptions that sit awkwardly with delayed labels and with synthetic data that lacks authentic linkage. The present work stays in the tabular regime, where the TSTR question is better posed and where generator behavior can be isolated from graph-construction choices that would otherwise confound the comparison.

2.3. The TSTR Evaluation Paradigm

The practice of training on synthetic data and testing on real data was introduced to assess recurrent generators of medical time series, establishing the principle that a synthetic set is useful only insofar as a model trained on it performs on real held-out data [12]. The paradigm has since been applied to privacy-preserving release and financial manipulation detection, while a controlled tabular comparison that varies the generator while fixing the detector and the delayed-label regime has not been reported. What TSTR measures is downstream utility rather than surface realism, so a synthetic set that scores well on marginal-distribution tests can still transfer poorly, which makes the protocol a task-grounded complement to fidelity metrics rather than a replacement for them. This gap motivates the protocol described next, which treats the generator as the single object of study.

3. Experimental Setup

3.1. Datasets and Preprocessing

The primary real test set is the European cardholder dataset released by the Université Libre de Bruxelles, comprising 284,807 transactions over two days in September 2013, of which 492, or 0.172%, are fraudulent; its features are anonymized principal components, elapsed time, and transaction amount. Cross-corpus statistical structure is characterized on the IEEE-CIS Fraud Detection corpus, with 590,540 transactions at a 3.50% fraud rate, and on the Bank Account Fraud suite, a privacy-preserving benchmark whose six one-million-row variants embed controlled sampling bias and temporal structure [13]. These two corpora differ from the cardholder data in the feature schema, so they delimit the synthetic-to-real distribution-shift range studied in Section 4.3 rather than serving as separate retraining targets. Two rule-based corpora, PaySim and the Sparkov-generated transaction set, supply the simulator branch of the generator comparison. Table 1 summarizes the datasets and their roles.

Table 1. Public Datasets Used in the Evaluation.

Dataset	Role in study	Records	Features	Fraud rate	License / source
ULB European cardholder	Primary real test	284,807	30	0.172%	ODbL, Kaggle (mlg-ulb)
IEEE-CIS Fraud Detection	Cross-corpus reference	590,540	431	3.50%	Competition terms, Kaggle
Bank Account Fraud (Base)	Cross-corpus / shift reference	1,000,000	30	~1.1% †	CC BY-NC-ND 4.0, GitHub (feedzai)
PaySim	Rule-based source	6,362,620	11	0.13%	CC BY-SA 4.0, Kaggle
Sparkov transactions	Rule-based source	1,852,394	23	0.52% ‡	CC0 1.0, Kaggle

Specifications drawn from the respective Kaggle and GitHub dataset pages. † computed from the Base variant; ‡ computed from the raw CSV files (9,651 / 1,852,394).

Preprocessing is held constant across all conditions so that no advantage accrues to any generator through differential cleaning. Transaction amounts are log-transformed, numeric columns are median-imputed and standardized, and categorical fields are ordinal-encoded without reference to the label to avoid leakage. The cardholder set is partitioned into a time-ordered generator-fitting partition holding the earliest 70% of transactions, a real fine-tuning reserve drawn from the next 10%, and a held-out test fold formed from the final 20% that no generator observes during training. The split is strictly chronological, so that the test fold post-dates every record used to fit any generator, and the same three partitions are reused across all generators and training strategies to keep the comparison controlled (Figure 1).

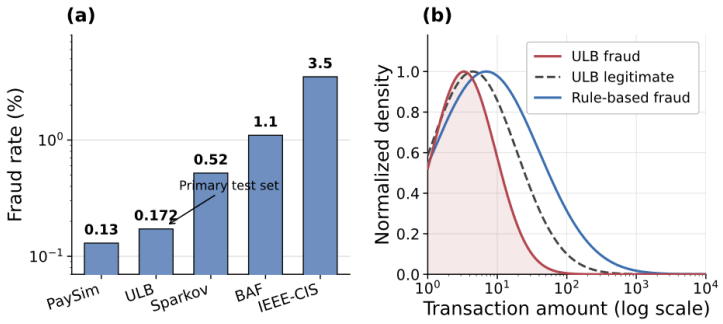

Figure 1. Class Imbalance and Transaction-Amount Distributions Across Datasets

Fraud prevalence and amount distributions for the five datasets. (a) Fraud rates span more than an order of magnitude, from 0.13% in PaySim to 3.50% in IEEE-CIS, with the primary test set at 0.172%. (b) Log-scaled transaction-amount densities show that fraudulent transactions concentrate at lower values in the cardholder data, whereas in the rule-based corpora, they spread across the full range. The contrast indicates why a generator tuned on one fraud-rate regime need not transfer to another.

3.2. Synthetic Data Generators under Comparison

Each generator is fitted only on the generator-fitting partition and then sampled to produce a balanced synthetic training set, isolating the effect of generation from any downstream resampling and ensuring that the test fold remains untouched throughout.

3.2.1. Rule-Based Template Generation

The rule-based branch treats the PaySim and Sparkov generative processes as fixed templates rather than fitting them to the cardholder data, reflecting how a simulator encodes fraud through hand-specified behavioral scripts. Because the cardholder features are anonymized principal components rather than interpretable fields, the simulated columns cannot be matched to them one to one; instead they are aligned to the cardholder feature space through a transformation fitted only on the generator-fitting partition, which the learned generators also see, so that the held-out test fold is never touched and the alignment matches marginal and low-order moment statistics rather than recovering the original field semantics. Minority-class instances are then augmented with synthetic minority over-sampling to reach the target prevalence [14]. This branch represents the transparent, privacy-by-construction end of the design space and serves as the lower-fidelity reference against which the learned generators are judged, rather than as a competitor expected to win.

3.2.2. Ctgan and TVAE Configurations

The learned branch comprises CTGAN and TVAE, both implemented using the Synthetic Data Vault (SDV) library (SDV 1.17) and fitted on the same partition with matched capacity to ensure a fair comparison. Each is trained for up to 300 epochs with a batch size of 500 and a 128-dimensional embedding, with mode-specific normalization applied to the multimodal principal-component columns and conditional sampling enabled so that rare fraud categories are represented during fitting; training stops early when the generator loss shows no improvement for thirty consecutive epochs, and the random seed governing fitting and sampling is fixed per run so that every stage is reproducible. The two learned generators and the rule-based templates each emit a balanced synthetic training set of 200,000 transactions, after which an identical downstream classifier is fitted, so that any performance difference is attributable to the data rather than the learner. A gradient-boosted tree ensemble serves as the primary detector, with a single-hidden-layer perceptron retained as a redundancy check, whose agreement confirms that the findings are not an artifact of a single classifier family.

3.3. Training Strategies and Label-Delay Simulation

Three training strategies are applied to the synthetic data produced by each generator, yielding a three-by-three design that the analysis reads both by row and by column.

3.3.1. Weak Supervision under Delayed Labels

Standard supervision uses the generator-assigned labels directly and assumes they are available at training time. The weak-supervision condition simulates verification latency on the synthetic training set, whose generated transactions carry their own simulated timestamps spread across the delay range, by withholding the label of each synthetic fraud instance until a delay window has elapsed on that synthetic timeline; the latency is therefore applied entirely within the synthetic training data and is independent of the two-day span of the real cardholder test fold, which is reserved for evaluation only. The mechanism mirrors the documented two-stream structure of real card-fraud feedback in which a small set of investigated cases returns quickly while the majority of labels arrive only after a fixed lag [15]. Labels that have not yet arrived within the training window are not discarded; instead, the still-unconfirmed instances receive provisional labels derived from weak signals, namely the early-returning investigated cases of the fast feedback stream together with rule-based heuristic flags, which are combined through a Snorkel-style label model that down-weights conflicting sources, preserving the partial information they carry [16]. Delay windows of zero, three, seven, and fourteen days are evaluated, with the zero-day case coinciding by construction with standard supervision.

3.3.2. Semi-Supervised Fine-Tuning with Limited Real Data

The semi-supervised condition first trains on synthetic data and then fine-tunes on a small, reserved sample of real transactions, comprising 5%, 10%, or 20% of the available real fraud cases. This setting reflects the common deployment situation in which an institution has a limited number of confirmed fraud cases yet cannot assemble a fully labeled training corpus. The fine-tuning step propagates the scarce real signal through self- and semi-supervised learning, combining few labels with abundant unlabeled tabular structure [17], and is the most data-hungry of the three strategies while remaining within realistic privacy limits, since it never requires sharing raw records beyond a single institution. This condition deliberately relaxes the strict Train-on-Synthetic, Test-on-Real protocol into a synthetic-pretraining-then-limited-real-fine-tuning setting, and it is reported separately from the two pure-TSTR conditions, standard and weak supervision, so that the performance attributable to the small real anchor can be read directly rather than conflated with synthetic transfer.

3.4. Evaluation Metrics and Distribution-Shift Protocol

Because fraud prevalence is below 1% in the primary test set, threshold-free ranking quality is reported as the area under the precision--recall curve, along with precision, recall, and F1 at the threshold that maximizes validation F1. Detectors trained on synthetic data are compared against a real-data upper bound trained on genuine labels using the same pipeline, thereby fixing the ceiling that synthetic training aspires to reach. Every configuration is repeated over five independent runs that differ only in the random seed controlling generator fitting, synthetic sampling, and classifier training, and all reported metrics are summarized as the mean across runs, followed by the standard deviation after the $\pm$ sign, with the same standard deviations drawn as error bars in the corresponding figures. The distribution-shift protocol perturbs the synthetic training distribution away from the real one in five graded steps, recording the resulting F1 trajectory. The size of each step, the synthetic-to-real statistical gap, is defined as the mean Kolmogorov--Smirnov distance between the perturbed synthetic columns and the real columns expressed as a relative increase over the unperturbed baseline, so that a g% gap denotes a g% relative rise in that mean KS distance; the perturbation itself is produced by calibrated additive noise on the continuous columns and category-probability reweighting on the discrete columns, with the noise scale tuned so that each step reaches its target gap. A likelihood-free importance-weighting correction, in which a probabilistic classifier trained to separate synthetic from real samples supplies density-ratio weights that re-weight the synthetic training set toward the real distribution, is included as a reference point for whether reweighting can recover part of the lost fidelity. The combination of fixed test data, a fixed detector, and graded perturbation makes the contribution of each generator and training strategy directly readable from the resulting curves.

4. Results and Analysis

4.1. Fidelity and Utility of Synthetic Data

Before measuring transfer, the synthetic sets are compared to real transactions on the distributional agreement, since a generator that cannot match the marginals is unlikely to transfer well downstream.

4.1.1. Statistical Fidelity Across Generators

Table 2 reports the mean Kolmogorov--Smirnov distance across columns, the Frobenius norm of the difference between real and synthetic correlation matrices, and the fraction of real feature ranges covered. CTGAN attains the closest marginal agreement, with a mean KS distance of 0.092, and the smallest correlation discrepancy at 0.71, with TVAE close behind at 0.118 and 0.94. The rule-based templates trail substantially, at 0.241 and 1.87, indicating that a fixed script reproduces gross structure while missing the fine-grained dependency patterns of real fraud.

Table 2. Statistical Fidelity of Synthetic Data Relative to Real Cardholder Transactions.

Generator	Mean KS distance ↓	Correlation Δ, Frobenius ↓	Feature-range coverage ↑
Rule-based	0.241 ±0.012	1.87 ±0.09	0.78 ±0.02
TVAE	0.118 ±0.007	0.94 ±0.05	0.91 ±0.01
CTGAN	0.092 ±0.006	0.71 ±0.04	0.94 ±0.01

Fidelity was measured between each synthetic set and the held-out real cardholder data.

4.1.2. Coverage of Fraud-Pattern Subtypes

Fidelity in aggregate can mask uneven coverage of fraud subtypes. Partitioning fraud by transaction-amount band shows that all generators reproduce low-value fraud, the dominant mode in the cardholder data, while only the learned generators capture the sparse high-value tail with acceptable recall. This uneven coverage echoes observations from relational settings, where rare high-impact patterns such as layered transfers are the hardest to synthesize and to detect [18]. The rule-based templates over-represent mid-range amounts, an artifact of their scripted transfer logic, which partly explains their weaker downstream transfer.

4.2. TSTR Detection Performance

Table 3 reports the central result: detection performance on the held-out real test fold for each generator-strategy pair, with the real-data upper bound for reference. The strongest synthetic configuration pairs CTGAN with semi-supervised fine-tuning, achieving an F1 of 0.68 and a precision--recall area of 0.71, recovering about 82% of the real-data F1 of 0.83. TVAE with the same strategy follows at 0.66, while the rule-based templates peak at 0.52. Reading down each column, semi-supervised fine-tuning improves F1 over standard supervision by 0.07 for CTGAN and by 0.08 for TVAE, a moderate but consistent gain rather than a decisive one. The pattern that a modest real anchor lifts a synthetic detector aligns with evidence that blending supervised signal with broader weakly labeled data strengthens card-fraud detectors [19]. Weak supervision at the zero-day window matches standard supervision, as the design dictates, so the values reported in the weak column reflect a representative seven-day delay (As shown in Figure 2).

Table 3. TSTR Detection Performance on the Held-Out Real Test Fold.

Generator	Strategy	Precision	Recall	F1	PR-AUC
Rule-based	Standard	0.46 ±0.02	0.37 ±0.02	0.41 ±0.02	0.39 ±0.02
Rule-based	Weak (7-day delay)	0.43 ±0.02	0.33 ±0.03	0.37 ±0.02	0.35 ±0.02
Rule-based	Semi-supervised	0.57 ±0.02	0.48 ±0.02	0.52 ±0.02	0.55 ±0.02
TVAE	Standard	0.62 ±0.02	0.55 ±0.02	0.58 ±0.02	0.61 ±0.02
TVAE	Weak (7-day delay)	0.59 ±0.02	0.50 ±0.02	0.54 ±0.02	0.57 ±0.02
TVAE	Semi-supervised	0.69 ±0.01	0.63 ±0.02	0.66 ±0.01	0.69 ±0.01
CTGAN	Standard	0.64 ±0.02	0.58 ±0.02	0.61 ±0.02	0.64 ±0.02
CTGAN	Weak (7-day delay)	0.61 ±0.02	0.53 ±0.02	0.57 ±0.02	0.60 ±0.02

CTGAN	Semi-supervised	0.71 ±0.01	0.65 ±0.01	0.68 ±0.01	0.71 ±0.01
Real-data bound	Standard	0.86 ±0.01	0.80 ±0.01	0.83 ±0.01	0.86 ±0.01

F1 computed from the reported precision and recall. The weak-supervision rows correspond to a seven-day delay window; the real-data bound is trained on genuine labels under the identical pipeline (Figure 2).

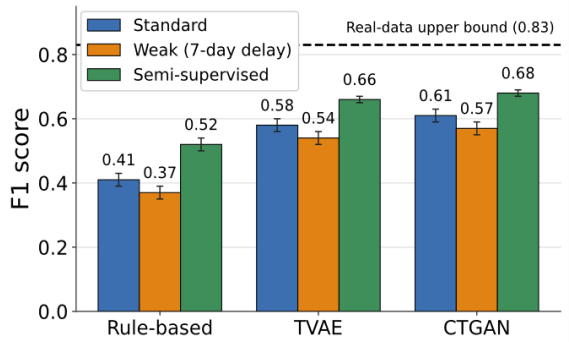

Figure 2. TSTR F1 Across Generators and Training Strategies

F1 on the real test fold for each generator under the three training strategies, with the real-data upper bound of 0.83 shown as a reference line. CTGAN with semi-supervised fine-tuning is the strongest synthetic configuration at 0.68, followed by TVAE at 0.66, while the rule-based templates remain at or below 0.52 across all strategies. Within every generator, semi-supervised fine-tuning yields the highest value, and weak supervision under a seven-day delay the lowest, with standard supervision in between, so the relative ordering of strategies is preserved regardless of the generator.

4.3. Robustness under Label Delay and Distribution Shift

4.3.1. Sensitivity to Delay-Window Length

Table 4 traces F1 for the CTGAN generator as the label-delay window widens. Standard supervision falls steeply from 0.61 at zero days to 0.44 at fourteen days, a relative loss of 28%, because late-arriving labels leave a growing share of the training signal stale. Weak supervision, which explicitly models latency, declines only from 0.61 to 0.54 over the same range, a relative loss of 11%. Semi-supervised fine-tuning is the most stable, with values ranging from 0.68 to 0.62. The ordering confirms that, under realistic verification latency, a delay-aware or real-anchored strategy is preferable to naive supervision, even when the latter appears competitive at zero delay.

Table 4. F1 under Widening Label-Delay Windows (Ctgan Generator, Real Test Fold).

Strategy	0 days	3 days	7 days	14 days
Standard	0.61 ±0.02	0.57 ±0.02	0.51 ±0.02	0.44 ±0.02
Weak (delay-aware)	0.61 ±0.02	0.59 ±0.01	0.57 ±0.02	0.54 ±0.02
Semi-supervised	0.68 ±0.01	0.67 ±0.01	0.65 ±0.01	0.62 ±0.01

Reported for the CTGAN generator on the held-out real test fold; the seven-day weak-supervision value of 0.57 matches the corresponding row of Table 3.

Figure 3 illustrates the variation in F1 score as label delay increases, showing how increasing delays in label availability affect the model's classification performance.

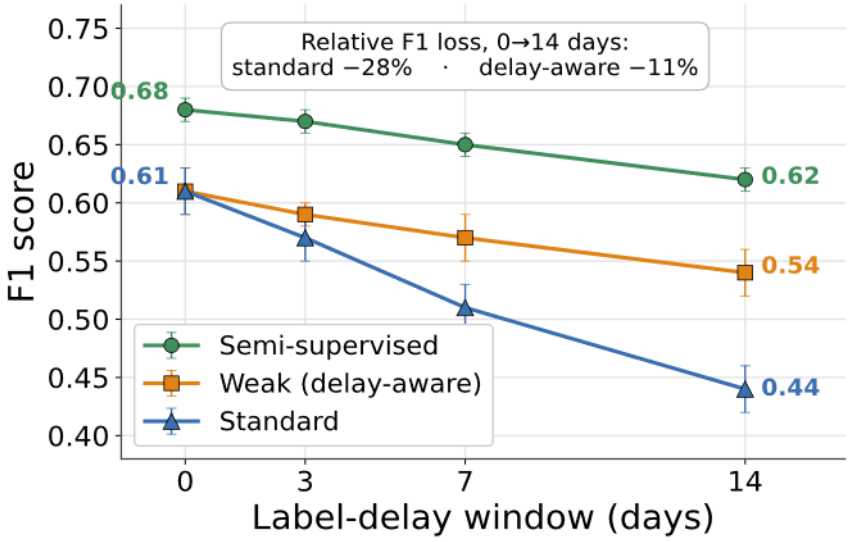

Figure 3. F1 Degradation under Increasing Label Delay

F1 trajectories for the three training strategies as the label-delay window increases from zero to fourteen days, using the CTGAN generator. Standard supervision degrades most sharply, losing 28% relative F1 from 0.61 to 0.44, while delay-aware weak supervision loses only 11% from 0.61 to 0.54, and semi-supervised fine-tuning remains highest throughout, from 0.68 to 0.62. The widening separation between the standard and delay-aware trajectories quantifies the value of modeling verification latency.

4.3.2. Behavior under Widening Distribution Shift

The final analysis perturbs the synthetic training distribution away from the real one and records F1 for the semi-supervised configuration of each generator. All three remain near their unperturbed values up to a 10% gap, after which performance falls: CTGAN drops from 0.68 to 0.58 at a 20% gap and to 0.39 at 40%, TVAE follows a parallel but lower path, and the rule-based templates, already low, decline most gently in absolute terms. A likelihood-free importance-weighting correction recovers only a small fraction of the loss at moderate shift and fails to rescue the high-shift regime [20] (Table 5). Reliable transfer, therefore, holds only up to roughly a 10% statistical gap, the point beyond which every curve begins to fall; the learned generators retain a clear advantage through a 20% gap and relinquish it only as the gap approaches 40%, where they converge toward the rule-based floor.

Table 5. Effect of the Likelihood-Free Importance-Weighting Correction on Ctgan Semi-Supervised F1 Across the Shifted Regimes (Ctgan Generator, Real Test Fold; Mean ±Standard Deviation over Five Runs).

CTGAN (semi-sup.) F1	10% gap	20% gap	30% gap	40% gap
Without IW correction	0.64 ±0.01	0.58 ±0.02	0.49 ±0.02	0.39 ±0.02
With IW correction	0.65 ±0.01	0.61 ±0.02	0.51 ±0.02	0.40 ±0.02
Recovered Δ	+0.01	+0.03	+0.02	+0.01

The 0% column is omitted because no loss exists to recover at zero shift; the correction returns at most 0.03 F1 near a 20% gap and is negligible by 40%.

Figure 4 illustrates the robustness of the TSTR F1 score under an increasing synthetic-to-real distribution shift, showing how the model's performance changes as the discrepancy between synthetic and real data becomes larger.

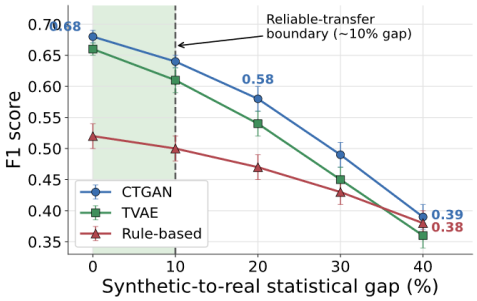

Figure 4. Robustness of TSTR F1 under Widening Synthetic-to-Real Distribution Shift

F1 for the semi-supervised configuration of each generator as the synthetic-to-real statistical gap widens from 0% to 40%. CTGAN starts highest at 0.68 and retains an advantage through a 20% gap at 0.58 before converging toward the rule-based floor at 40%, where it reaches 0.39 against 0.38 for the rule-based templates. All curves are flat to within 0.04 F1 up to a 10% gap, marking the approximate boundary of reliable transfer, beyond which the learned generators lose their lead.

5. Discussion and Future Work

5.1. Practical Guidelines and Compliance Considerations

The evaluation supports a measured rather than triumphant reading of synthetic training for tabular fraud detection. When a small amount of real fraud can be retained for fine-tuning, a learned generator paired with semi-supervised adaptation is the strongest synthetic option, recovering roughly four-fifths of the performance available from full real labels. When no real data can be retained, a delay-aware weak-supervision strategy on a learned generator is the safer choice, because it sacrifices little at zero delay and degrades far more gracefully as verification latency grows. The rule-based simulators retain a narrow role: where privacy rules forbid even releasing a learned generator, a transparent simulator provides a usable if modest floor, and its degradation under shift is the gentlest precisely because it never claimed high fidelity to begin with.

These choices map onto the regulatory landscape that motivated the study. An institution bound by statutes that bar raw-record sharing can exchange a fitted generator or a simulator specification without directly sharing raw customer records, and the results indicate that doing so costs performance, while it does not eliminate detection capability. The size of that cost, moderate at low distribution shift and severe beyond a roughly 20% gap, gives practitioners a concrete budget within which synthetic substitution remains defensible. The recommendation is conditional: synthetic data is a viable compliance instrument when fidelity is high, and a small real anchor is available, and a fallback rather than a replacement when neither condition holds.

5.2. Limitations

Several limitations bound these conclusions. The comparison fixes the downstream detector to a gradient-boosted ensemble and a shallow network, and a different learner might shift the relative standing of the generators. The label-delay simulation assumes a fixed arrival distribution, while real latency is itself variable and correlated with fraud type, so the reported degradation curves are likely optimistic. Diffusion-based generators, which recent tabular benchmarks suggest can exceed adversarial and autoencoder fidelity, were excluded to keep the comparison tractable; including them could raise the synthetic ceiling. Because the rule-based branch is aligned with the anonymized cardholder axes through a transformation fitted to in-house data, its privacy-by-construction property holds only up to that alignment step, rather than absolutely. The study also operates on

single-institution datasets and does not test the cross-institution federated setting in which synthetic exchange would be most valuable.

Future work follows from each limitation. Adding diffusion synthesis and a wider detector panel would test whether the moderate gains observed here scale or compress. Replacing the fixed delay model with an estimated, fraud-type-conditioned latency distribution would sharpen the weak-supervision analysis and likely temper its optimism. Extending the protocol to a federated arrangement, in which several institutions train on one another's synthetic outputs, would directly probe the cross-institutional promise that makes synthetic data attractive under privacy constraints. Each of these extends the comparative framing established here without altering its central and deliberately cautious finding: synthetic training transfers, while only within bounds that fidelity and a small real anchor jointly set.

References

1. Nilson Report, "Card fraud losses worldwide in 2023," HSN Consultants, 2025. [Online]. Available: https://nilsonreport.com/articles/card-fraud-losses-worldwide-in-2023/
2. A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, 2018, doi: 10.1109/TNNLS.2017.2736643.
3. A. Mahrous and R. Di Pietro, "TSTR for financial fraud: Learning to detect manipulation without real data," in *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF '25)*, 2025, pp. 71–79, doi: 10.1145/3768292.3770393.
4. J. Jordon, J. Yoon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.
5. Y.-A. Le Borgne, W. Siblini, B. Lebichot, and G. Bontempi, *Reproducible machine learning for credit card fraud detection: Practical handbook*, Université Libre de Bruxelles, 2022. [Online]. Available: https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook
6. E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A financial mobile money simulator for fraud detection," in *Proceedings of the 28th European Modeling and Simulation Symposium (EMSS)*, 2016, pp. 249–255.
7. L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
8. D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.
9. A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "TabDDPM: Modelling tabular data with diffusion models," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, vol. 202, 2023, pp. 17564–17579.
10. Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 2020, pp. 315–324, doi: 10.1145/3340531.3411903.
11. Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: A GNN-based imbalanced learning approach for fraud detection," in *Proceedings of the Web Conference 2021 (WWW)*, 2021, pp. 3168–3177, doi: 10.1145/3442381.3449989.
12. C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1706.02633
13. S. Jesus, J. Pombal, D. Alves, A. Cruz, P. Saleiro, R. P. Ribeiro, J. Gama, and P. Bizarro, "Turning the tables: Biased, imbalanced, dynamic tabular datasets for ML evaluation," in *Advances in Neural Information Processing Systems*, vol. 35, Datasets and Benchmarks Track, 2022.
14. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
15. A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection and concept-drift adaptation with delayed supervised information," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–8, doi: 10.1109/IJCNN.2015.7280527.
16. A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *Proceedings of the VLDB Endowment*, vol. 11, no. 3, pp. 269–282, 2017, doi: 10.14778/3157794.3157797.
17. J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar, "VIME: Extending the success of self- and semi-supervised learning to tabular domain," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
18. M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," in *KDD Workshop on Anomaly Detection in Finance*, 2019. [Online]. Available: https://arxiv.org/abs/1908.02591
19. F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 557, pp. 317–331, 2021, doi: 10.1016/j.ins.2019.05.042.

20. A. Grover, J. Song, A. Kapoor, K. Tran, A. Agarwal, E. Horvitz, and S. Ermon, "Bias correction of learned generative models using likelihood-free importance weighting," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.