Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Synthetic Data Generation in AI and Its Applications

Shivam Kumar[1],Isha[2], Dr. Vishal Shrivastava[3], Dr. Akhil Pandey[4]
[1,2,3,4]Artificial Intelligence & Data Science, Arya College of Engineering & I.T. Jaipur, India

Abstract

Modern artificial intelligence systems depend on large, diverse and well-labelled datasets, yet real data is often scarce, imbalanced, expensive to annotate, or restricted by privacy regulation. Synthetic data generation (SDG) addresses these limits by producing artificial records that preserve the statistical structure of real data without copying individual records. This paper reviews the main families of SDG methods — simulation, statistical models, and deep generative models (GANs, VAEs, diffusion models) — and their applications in healthcare, finance, autonomous driving, cybersecurity and language modelling. It then evaluates four generators (Gaussian copula, Gaussian mixture model, SMOTE-style interpolation and a jittered bootstrap) on the Breast Cancer Wisconsin dataset using a unified fidelity–utility–privacy protocol averaged over ten random splits. Models trained only on synthetic data reached up to 95.56% test accuracy against 95.96% for real data. Under severe class imbalance, synthetic minority oversampling raised minority-class F1 from 85.55% to 94.01%. Interpolation and bootstrap generators, however, produced records far closer to real patients than the statistical generators, exposing a privacy trade-off. The results show that no single metric is sufficient to judge synthetic data.

Keywords: SYNTHETIC DATA, GENERATIVE MODELS, GANs, DATA AUGMENTATION, PRIVACY, FIDELITY, UTILITY, TABULAR DATA, HEALTHCARE, DIFFERENTIAL PRIVACY.

1. Introduction

Machine learning performance is closely tied to the quantity and quality of training data. In practice, the data that matters most is often the hardest to obtain: medical records are protected by privacy law, fraud and intrusion events are rare, and labelling images for autonomous driving costs time and money. Synthetic data generation (SDG) offers a way around these limits by creating artificial data that can stand in for, or supplement, real data [1].

Synthetic data is judged along three axes. **Fidelity** measures how closely the synthetic distribution matches the real one. **Utility** measures whether a model trained on synthetic data performs well on real data, usually through the Train-on-Synthetic, Test-on-Real (TSTR) protocol. **Privacy** measures whether the synthetic records leak information about real individuals. These goals pull against one another: a generator that copies its training data has perfect fidelity and utility but no privacy.

This paper makes three contributions: (i) a structured review of SDG methods and applications (Tables 1–2); (ii) a reproducible evaluation protocol covering fidelity, utility and privacy (Table 3); and (iii) an experiment comparing four generators and testing synthetic augmentation under class imbalance and data scarcity. Section 2 reviews related work, Section 3 presents the methodology, Section 4 reports results, and Section 5 concludes.

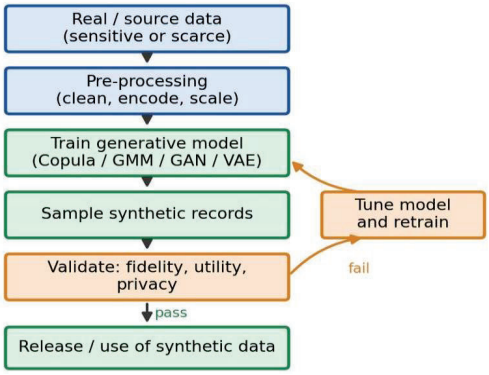


Fig. 1: General synthetic data generation pipeline

1.1 Families of Generation Methods

SDG methods fall into three groups (Fig. 2). Simulation-based methods render data from physical or rule-based models, as in driving simulators and domain randomization [2][3]. Statistical methods fit an explicit probability model — a copula, mixture model or Bayesian network — and sample from it [4][5]. Deep generative models learn the distribution with neural networks and currently dominate image, text and time-series generation.

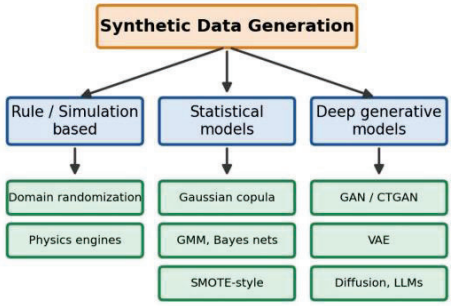


Fig. 2: Taxonomy of synthetic data generation methods

1.2 Generative Adversarial Networks

A GAN trains a generator G to map random noise to realistic samples while a discriminator D learns to separate real from generated samples (Fig. 3). The two networks play a min-max game that, at equilibrium, drives the generated distribution toward the real one [6]. Conditional variants such as CTGAN handle mixed discrete and continuous columns in tabular data [7]. GANs can suffer from mode collapse and unstable training.

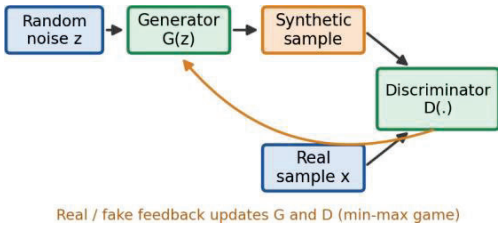


Fig. 3: Generator–discriminator structure of a GAN

1.3 VAEs, Diffusion Models and Privacy

Variational autoencoders learn a latent space and decode samples from it, giving stable training but sometimes blurry or over-smoothed output [8]. Diffusion models generate data by reversing a gradual noising process and now produce the most realistic images

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

[9]. Because any generator can memorize its training set, formal protection such as differential privacy (DP) is often added; PATE-GAN, for example, trains the discriminator with a DP mechanism [10][11].

Table 1: Summary of Synthetic Data Generation Methods

Method	Category	Principle	Strength	Limitation
Simulation / rendering	Rule-based	Render data from physical or rule-based models	Perfect labels; rare events on demand	Reality gap between simulated and real data
Gaussian copula / GMM	Statistical	Fit marginals and dependence, then sample	Fast, interpretable, little data needed	Weak on complex, high-dimensional structure
SMOTE-style interpolation	Statistical	Interpolate between nearest neighbours	Simple; good for minority oversampling	Stays close to real points; privacy risk
GAN / CTGAN	Deep generative	Adversarial generator–discriminator game	Sharp, realistic samples	Unstable training; mode collapse
VAE	Deep generative	Encode to latent space, decode samples	Stable training; latent control	Over-smoothed output
Diffusion models	Deep generative	Reverse a gradual noising process	State-of-the-art image quality	High compute; slow sampling

Table 2: Applications of Synthetic Data Across Domains

Domain	Typical use	Data type	Key benefit	Main concern
Healthcare	Share patient-like records for research and model development	Tabular, imaging, time series	Privacy-friendly data sharing	Re-identification; clinical realism
Finance	Fraud detection with rare fraud cases; stress testing	Tabular transactions	Balances rare-event classes	Hidden bias; regulatory approval
Autonomous driving	Simulated roads, weather and accident scenarios	Images, LiDAR	Safe coverage of dangerous cases	Sim-to-real gap
Cybersecurity	Attack traffic for intrusion detection	Network logs	Training without exposing real logs	Attackers may adapt
NLP / LLMs	Instruction and dialogue data for fine-tuning	Text	Cheap, scalable labelling	Model collapse if recycled
Software testing	Test databases that mimic production	Tabular, relational	No production data in test systems	Referential integrity

Table 3: Evaluation Metrics Used for Synthetic Data

Dimension	Metric	What it measures	Desired value
Fidelity	Mean Kolmogorov–Smirnov (KS) statistic	Per-column distance between real and synthetic marginals	Lower
Fidelity	Correlation difference	Mean absolute gap between correlation matrices	Lower
Utility	TSTR accuracy and F1	Real-data performance of a model trained on synthetic data	Close to real-data baseline
Privacy	Distance to closest record (DCR)	Median distance from each synthetic row to nearest real training row	Close to hold-out level or higher
Privacy	Exact-match rate	Share of synthetic rows duplicating a real row	0%

2. Related Works

2.1 Statistical and Interpolation Approaches

Rubin proposed releasing multiply-imputed synthetic microdata as a way to limit disclosure in official statistics [4]. SMOTE generates minority-class examples by interpolating between a sample and its nearest neighbours and remains a standard remedy for imbalance [12]. The Synthetic Data Vault modelled relational databases with copulas and hierarchical statistical models [5].

2.2 Adversarial and Deep Generative Approaches

Goodfellow et al. introduced GANs [6]. medGAN adapted them to discrete patient records [13], and recurrent conditional GANs were proposed for medical time series [14]. CTGAN added mode-specific normalization and conditional sampling for tabular data and reported gains over Bayesian-network baselines [7]. VAEs [8] and diffusion models [9] offer more stable training at higher computational cost.

2.3 Privacy-Preserving Generation

Differential privacy gives a mathematical bound on what any output can reveal about one individual [11]. PATE-GAN applies it to GAN training [10]. Surveys stress that fidelity, utility and privacy must be reported together because improving one can degrade another [1].

2.4 Simulation-Based Data

Domain randomization trains vision models on randomized simulated scenes that transfer to real robots [2], and open simulators such as CARLA provide labelled driving data at scale [3]. Their shared weakness is the sim-to-real gap.

3. Proposed Methodology

This study compares four generators under one protocol on a real tabular dataset, then tests synthetic augmentation in two data-limited settings.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

3.1 Dataset and Generators

The Breast Cancer Wisconsin (Diagnostic) dataset [15], loaded through scikit-learn [16], has 569 samples, 30 numeric features and two classes (212 malignant, 357 benign). Each generator is fitted separately to each class of the training data and sampled to match the real class counts. **Gaussian copula:** rank-transformed marginals with a multivariate normal dependence structure. **GMM:** a three-component full-covariance Gaussian mixture. **SMOTE-style:** interpolation between a point and one of its five nearest neighbours. **Jittered bootstrap:** resampled real rows with Gaussian noise at 10% of each feature's standard deviation, used as a naive baseline.

3.2 Evaluation Protocol

Data is split 70/30 with stratification and standardized using training statistics. Fidelity, utility and privacy are measured as in Table 3. A random forest of 200 trees is the downstream classifier. Every figure reported is the mean over ten random splits.

3.3 Proposed Algorithm

Input: real dataset *D*. **Output:** fidelity, utility and privacy scores per generator.

1. Split *D* into training and test sets (70/30, stratified) and standardize.
2. Fit each generator on the real training data of each class.
3. Sample synthetic data with the same class counts as the training set.
4. Compute KS statistic, correlation difference, DCR and exact matches.
5. Train on synthetic data, test on real data; record accuracy and F1.
6. Augmentation tests: add synthetic minority samples to an imbalanced set; vary the number of real samples.
7. Repeat for ten random seeds and report mean ± standard deviation.

3.4 Proposed Flowchart

Fig. 4 shows the end-to-end experimental flow, from loading the data to comparing every generator against the real-data baseline.

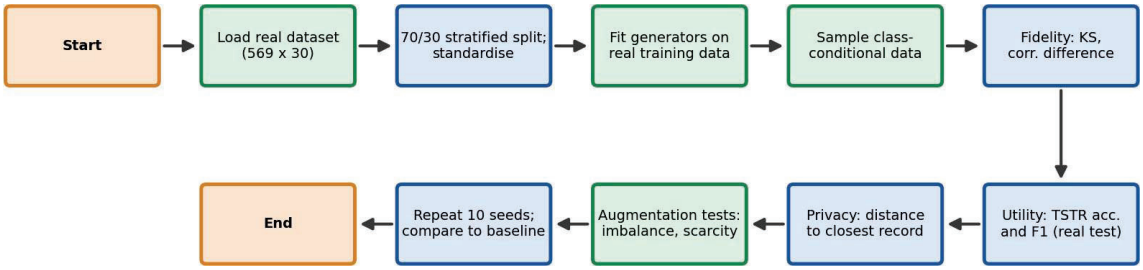


Fig. 4: Experimental methodology flowchart

4. Results and Discussion

All experiments used Python with scikit-learn [16]. Each configuration was repeated over ten random stratified splits, and the real-data baseline (TRTR, train and test on real data) reaches 95.96% accuracy and 96.82% F1. Table 4 summarizes fidelity, utility and privacy for the four generators.

Table 4: Real vs. Synthetic Data Performance (mean over 10 runs)

Training data	Accuracy (%)	F1 (%)	KS ↓	Corr. diff ↓	DCR	Exact match (%)
Real data (baseline)	95.96	96.82	—	—	2.14 (hold-out)	—
Gaussian Copula	95.56	96.45	0.040	0.038	2.22	0.0
GMM	95.50	96.41	0.058	0.033	2.32	0.0
SMOTE-style	94.97	96.00	0.054	0.043	0.62	0.0
Jittered Bootstrap	95.26	96.25	0.043	0.031	0.44	0.0

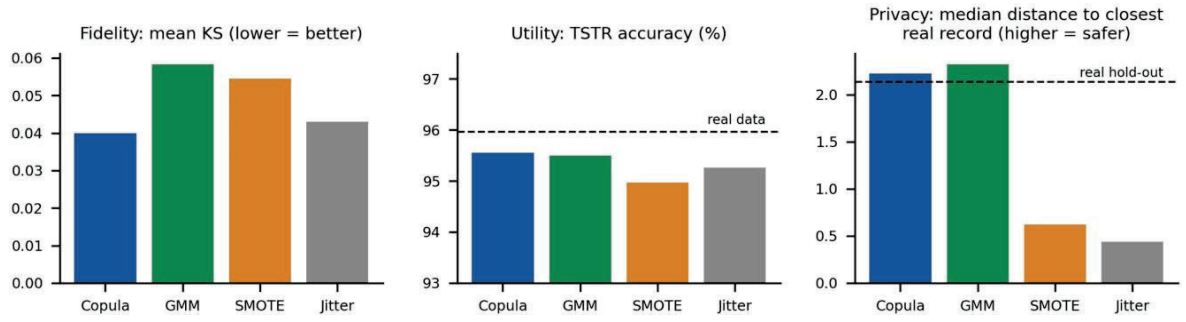


Fig. 5: Fidelity, utility and privacy of the four generators

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

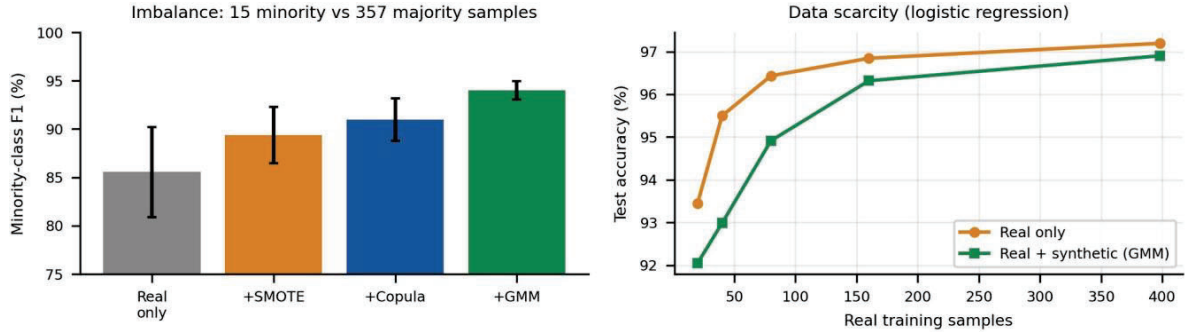


Fig. 6: Synthetic augmentation under class imbalance and data scarcity

4.1 Fidelity and Utility

All four generators reproduced the real marginals closely (mean KS between 0.040 and 0.058), and the Gaussian copula matched marginals best. Classifiers trained purely on synthetic data scored 94.97% to 95.56% accuracy, within about one percentage point of the 95.96% real-data baseline. This shows that, for a clean low-dimensional tabular task, simple statistical generators already capture most of the information a classifier needs.

4.2 Privacy

Utility and fidelity alone hid an important difference. The copula and GMM generators produced records at a median distance of 2.22 and 2.32 from the closest real training record, comparable to the 2.14 seen for genuinely unseen real patients. The SMOTE-style and jittered-bootstrap generators sat at 0.62 and 0.44, roughly three to five times closer, because they are built directly from real rows. No generator produced exact duplicates, so an exact-match check alone would wrongly report all four as safe.

4.3 Augmentation Under Imbalance and Scarcity

With only 15 malignant training samples against 357 benign, a real-only classifier reached 85.55% minority-class F1 (90.58% accuracy). Adding synthetic minority samples raised F1 to 89.37% (SMOTE-style), 91.00% (copula) and 94.01% (single Gaussian), the latter also reducing run-to-run variation. In the scarcity test the picture was less favorable: with a logistic-regression classifier, adding 600 synthetic samples from a single Gaussian slightly lowered accuracy at every size, for example 93.45% to 92.05% with 20 real samples and 97.19% to 96.90% with all 398. A generator fitted on very few samples inherits their estimation error, so synthetic data cannot add information that the real sample does not contain.

4.4 Discussion

Three findings stand out. First, synthetic data is most useful where the problem is a distribution imbalance rather than missing information. Second, the generator with the best fidelity (copula) was also among the safest, while the simplest oversampling methods were the most exposed; the two can therefore be traded off deliberately. Third, utility can be near-perfect while privacy is poor, which supports reporting all three metric groups together [1]. These findings are consistent with the review in Section 2.

5. Conclusion and Future Work

This paper reviewed synthetic data generation methods and applications and evaluated four generators on a unified fidelity–utility–privacy protocol. Models trained on synthetic data came within about one percentage point of real-data accuracy, and synthetic minority oversampling improved minority-class F1 from 85.5% to as high as 94.0% under severe imbalance. At the same time, interpolation-based generators stayed much closer to real records than statistical ones, and synthetic augmentation did not help when real data was merely small.

The study has limits. It uses a single small public dataset, classical generators rather than deep models, and distance-based privacy measures rather than formal guarantees or membership-inference attacks. Future work will add CTGAN, VAE and diffusion generators, evaluate several datasets including images and time series, apply differential-privacy training, and test attack-based privacy audits.

6. References

1. Jordon, J. et al. “Synthetic Data – what, why and how?” arXiv:2205.03257, 2022.
2. Tobin, J. et al. “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World.” *IEEE/RSJ IROS*, 2017, pp. 23–30.
3. Dosovitskiy, A. et al. “CARLA: An Open Urban Driving Simulator.” *Conference on Robot Learning (CoRL)*, PMLR 78, 2017, pp. 1–16.
4. Rubin, D. B. “Discussion: Statistical Disclosure Limitation.” *Journal of Official Statistics* 9(2), 1993, pp. 461–468.
5. Patki, N., Wedge, R., & Veeramachaneni, K. “The Synthetic Data Vault.” *IEEE DSAA*, 2016, pp. 399–410.
6. Goodfellow, I. et al. “Generative Adversarial Nets.” *Advances in Neural Information Processing Systems* 27, 2014, pp. 2672–2680.
7. Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. “Modeling Tabular Data using Conditional GAN.” *NeurIPS* 32, 2019.
8. Kingma, D. P., & Welling, M. “Auto-Encoding Variational Bayes.” *ICLR*, 2014 (arXiv:1312.6114).
9. Ho, J., Jain, A., & Abbeel, P. “Denoising Diffusion Probabilistic Models.” *NeurIPS* 33, 2020.
10. Jordon, J., Yoon, J., & van der Schaar, M. “PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees.” *ICLR*, 2019.
11. Dwork, C., & Roth, A. “The Algorithmic Foundations of Differential Privacy.” *Foundations and Trends in Theoretical Computer Science* 9(3–4), 2014, pp. 211–407.
12. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. “SMOTE: Synthetic Minority Over-sampling Technique.” *Journal of Artificial Intelligence Research* 16, 2002, pp. 321–357.
13. Choi, E. et al. “Generating Multi-label Discrete Patient Records using Generative Adversarial Networks.” *Machine Learning for Healthcare (MLHC)*, PMLR 68, 2017, pp. 286–305.
14. Esteban, C., Hyland, S. L., & Rätsch, G. “Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs.” arXiv:1706.02633, 2017.
15. Street, W. N., Wolberg, W. H., & Mangasarian, O. L. “Nuclear Feature Extraction for Breast Tumor Diagnosis.” *SPIE* 1905, 1993, pp. 861–870.
16. Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12, 2011, pp. 2825–2830.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)