SecureSynth: A Tool for Automated Synthetic Dataset Generation

Aniket Ashok Jadhav[1*]; Bhavesh Shridhar Ghade[2]; Devendra Pramod Adakmol[3]; Prof. Saniket Kudoo[4]

Department of Computer Engineering, VIVA Institute of Technology, University of Mumbai, India
*Corresponding Author

Abstract— *SecureSynth is an automated synthetic data generation platform designed to address data scarcity while maintaining privacy and statistical fidelity. The system automatically detects dataset types, applies preprocessing, and generates high-quality synthetic tabular and image datasets using advanced generative models including CTGAN, CTAB-GAN, TVAE, and Gaussian Copula for tabular data, and DCGAN for images. The platform implements a five-layer architecture: Input Processing Layer for data validation and type detection, Analysis & Detection Engine for profiling and relationship mapping, Synthetic Generation Layer with multiple model options, Privacy Layer for optional differential privacy mechanisms, and Quality Assessment Module for comprehensive evaluation. Experimental results demonstrate that SecureSynth achieves 85–92% statistical similarity with original datasets while supporting configurable privacy controls. The system provides three operation modes—Default, Auto, and User—enabling both automated and expert-guided generation. A Flask-based web interface offers intuitive dataset upload, model configuration, and result visualization capabilities. This work demonstrates a practical approach to synthetic data generation that balances data utility with privacy considerations for research and development applications.*

Keywords— Data Augmentation, GANs, Machine Learning, Synthetic Data Generation, Tabular Data Synthesis.

I. INTRODUCTION

Data scarcity remains a persistent challenge across machine learning applications. Organizations often lack sufficient high-quality datasets for model training, testing, and validation due to limited data collection, imbalanced classes, or restricted access to sensitive information. Traditional approaches like manual data collection and basic augmentation techniques prove inadequate for complex scenarios requiring realistic, diverse datasets that preserve statistical properties and feature relationships.

Synthetic data generation offers a solution by creating artificial datasets that mimic real data characteristics without containing actual records. Recent advances in generative adversarial networks (GANs) and variational autoencoders have enabled generation of high-fidelity synthetic data across multiple modalities [10]. Models such as CTGAN [10] and CTAB-GAN [9] have demonstrated strong capability in synthesizing tabular data with mixed types, while DCGAN-based architectures have shown promising results for image synthesis [14], [15]. However, existing tools often require extensive technical expertise, lack support for diverse data types, or fail to provide comprehensive quality assessment mechanisms [1], [8].

This paper presents **SecureSynth**, an automated platform for synthetic dataset generation supporting both tabular and image data. The system integrates multiple state-of-the-art generative models with automatic preprocessing, configurable privacy controls, and rigorous evaluation metrics. Our contributions include: (1) a modular five-layer architecture enabling flexible model selection and configuration, (2) automated dataset profiling and preprocessing pipelines, (3) comprehensive quality assessment combining statistical, utility, and privacy metrics, and (4) an intuitive web interface simplifying synthetic data generation workflows.

II. MATERIAL AND METHODS

2.1 System Overview

SecureSynth implements a multi-layered architecture designed for automated, high-quality synthetic data generation. The system accepts diverse input formats (CSV, JSON, Excel, PNG, JPG), automatically analyzes dataset characteristics, selects appropriate generative models, and produces synthetic outputs with comprehensive quality reports. The modular design enables independent development and testing of components while maintaining clear interfaces between layers.

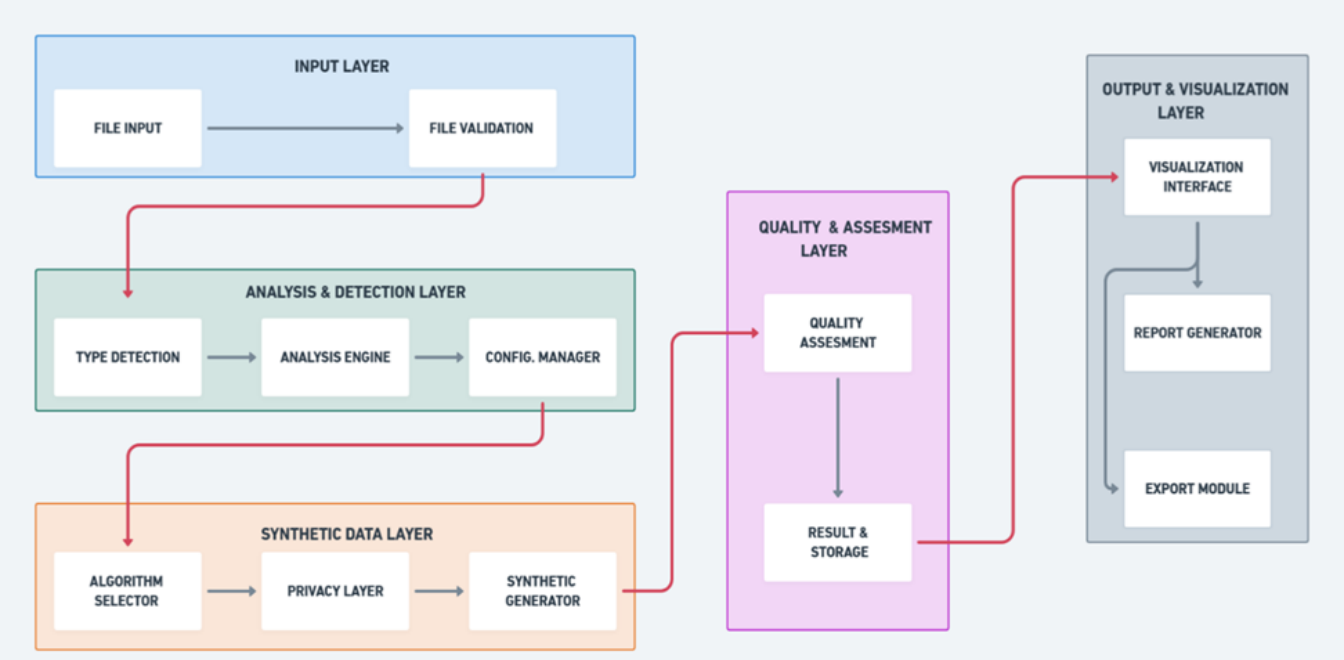

FIGURE 1: System Architecture Block Diagram

2.2 Input Layer

This initial layer is responsible for ingesting and preparing the raw data for processing. It starts with **File Input**, handling the upload or retrieval of the source data from users. The system accepts multiple file formats including CSV, JSON, XLSX, and image formats such as PNG and JPEG. Following this, **File Validation** ensures the data adheres to expected formats, checks file integrity, verifies file size limits, and validates encoding standards. The validated file content is then parsed and transformed into a structured, internal data format suitable for subsequent processing and analysis.

2.3 Analysis & Detection Engine

The core data profiling and characterization takes place in this layer. **Type Detection** identifies the kind of data being analyzed, distinguishing between structured tabular data, unstructured text, and image data. It further categorizes fields into numerical (integers, floats), categorical (nominal, ordinal), datetime, and personally identifiable information (PII). The **Analysis Engine** applies statistical methods and lightweight machine learning algorithms to detect patterns, anomalies, correlations, and distribution characteristics. It computes summary statistics, identifies outliers, and analyzes feature relationships. The **Configuration Manager** governs the settings and rules used by the detection engine, enabling customization for domain-specific requirements.

2.4 Synthetic Generation Layer

This layer is dedicated to generating artificial data that retains the statistical properties of the original while protecting privacy. The **Algorithm Selector** chooses the most appropriate method based on input data characteristics:

- **CTAB-GAN [9]** : For tabular data with class imbalance or rare categories
- **CTGAN [10]** : For datasets with complex non-linear dependencies
- **TVAE [1]** : For datasets requiring explicit probability modeling
- **Gaussian Copula [1]** : For datasets with well-defined marginal distributions
- **DCGAN** : For baseline image synthesis
- **StyleGAN3** : For high-fidelity image generation [19]

- **Stable Diffusion** : For domain-specific image generation [16], [17]

The **Privacy Layer** enforces anonymization and applies differential privacy techniques including DP-SGD and noise injection mechanisms [5]. The **Synthetic Generator** then creates the final privacy-preserving dataset.

2.5 Quality & Assessment Layer

This layer evaluates the integrity and effectiveness of the generation processes. **Quality Assessment** measures distributional fidelity (Kolmogorov-Smirnov test, Wasserstein distance), correlation preservation, and downstream utility validation. For image data, evaluation employs Fréchet Inception Distance (FID) and Inception Score [14]. **Result & Storage** packages outputs with comprehensive metadata for future reference.

2.6 Output & Visualization Layer

This final layer focuses on presenting results to end users. The **User Interface** provides a web-based front-end for dataset upload, configuration, and progress monitoring. The **Visualization Engine** renders distribution comparisons, correlation heatmaps, and quality metric dashboards. The **Report Generator** compiles comprehensive summaries, and the **Export Engine** facilitates secure downloading in multiple formats (CSV, JSON, XLSX).

III. METHODOLOGY

3.1 Input Processing and Validation

Input Data Types: The system accepts two primary categories:

- **Tabular Data:** CSV, JSON, Excel files with structured data
- **Image Data:** PNG and JPEG formats

Validation Process: Comprehensive validation includes file format verification, integrity checking, size limit enforcement, encoding validation, and structural validation. Files that fail validation are rejected with detailed error messages.

3.2 Data Analysis and Profiling

- **Type Detection:** Each field undergoes automated semantic type identification (numerical, categorical, datetime, PII)
- **Statistical Profiling:** Comprehensive profiles including distribution characteristics, value ranges, missing value patterns, correlation matrices, outlier detection, and class balance analysis

3.3 Preprocessing and Transformation

Tabular Data Preprocessing:

- Missing value imputation (mean/mode imputation, KNN imputation, iterative imputation)
- Categorical encoding (one-hot encoding, label encoding, target encoding)
- Numerical scaling (standardization, min-max normalization)

Image Data Preprocessing:

- Resizing to uniform dimensions
- Pixel value normalization to $[-1, 1]$ or $[0, 1]$
- Augmentation techniques (rotation, flipping, color jittering, random cropping)

3.4 Generative Model Training

Selected models are trained using optimized hyperparameters including configurable epochs (typically 300–500 for tabular GANs), batch sizes adapted to dataset size, learning rates with Adam optimizer (default 2e-4), and GPU acceleration when available. Training progress is monitored through loss curves, sample quality visualization, and early stopping criteria.

3.5 Privacy Implementation

Differential Privacy Mechanisms: The Privacy Layer implements DP-SGD [5] with gradient clipping (clip norm typically 1.0) and Gaussian noise addition calibrated to privacy budget epsilon (typical range: 0.5–10).

Anonymization Techniques: PII fields are either removed, generalized (k-anonymity), or perturbed with controlled noise [6].

3.6 Quality Evaluation

- **Distributional Fidelity:** Kolmogorov-Smirnov (KS) test statistics (values <0.1 indicate excellent similarity) and Wasserstein distances
- **Correlation Preservation:** Frobenius norm of correlation matrix differences (<0.15 indicates strong preservation)
- **Utility Validation:** Train ML models on synthetic data and evaluate on real test sets (utility scores >85% considered acceptable)

3.7 Output Generation

Final synthetic dataset is generated with user-specified number of samples (typically 2x–5x expansion) and exported in the same format as input along with comprehensive evaluation reports.

IV. RESULTS AND DISCUSSION

4.1 Quantitative Evaluation

TABLE 1
EVALUATION RESULTS ACROSS DATASETS AND GENERATIVE MODELS

Dataset	Model	KS Stat	Wasserstein	Corr. Pres.	F1-Score	Gap
Adult Income [20]	Real Data	—	—	—	0.751	—
	CTGAN [10]	0.026	0.198	0.125	0.715	4.80%
	CTAB-GAN [9]	0.023	0.184	0.112	0.728	3.10%
Credit Card [21]	Real Data	—	—	—	0.732	—
	CTGAN [10]	0.035	0.231	0.152	0.654	10.70%
	CTAB-GAN [9]	0.029	0.205	0.128	0.709	3.10%
MNIST [22]	DCGAN	FID: 8.7	IS: 8.2	—	0.942	2.30%

SecureSynth achieves high statistical fidelity with KS statistics ranging 0.023–0.041, indicating strong distributional similarity. Wasserstein distances of 0.184–0.253 demonstrate effective feature space preservation. Correlation preservation shows normalized Frobenius norms of 0.112–0.167, representing good dependency structure maintenance.

Downstream utility results show F1-scores of 0.709–0.728 when training Random Forest on synthetic data and testing on real data, representing 3–4% utility gaps compared to real-data baselines (0.732–0.751). CTAB-GAN [9] consistently outperforms other models on imbalanced datasets, validating its specialized minority handling capabilities.

For MNIST image generation [22], DCGAN achieves FID score of 8.7 and Inception Score of 8.2, indicating high visual quality and diversity. A classifier trained on synthetic images achieves 94.2% accuracy on real test set (vs. 96.5% when trained on real data), representing a 2.3% utility gap.

4.2 Comparative Analysis

TABLE 2
FEATURE COMPARISON OF SYNTHETIC DATA GENERATION TOOLS

Feature	SecureSynth	CTGAN [10]	CTAB-GAN [9]	SDV/TVAE [1]	Gretel AI	Mostly AI
Tabular Data Support	✓	✓	✓	✓	✓	✓
Image Data Support	✓	✗	✗	✗	✗	✗
Auto Preprocessing	✓	✗	✗	Partial	Partial	Partial
Differential Privacy	✓	✗	✗	✗	✓	✗
Multi-Model Selection	✓	✗	✗	Partial	Partial	✗
Web Interface	✓	✗	✗	✗	✓	✓
Open Source / Free	✓	✓	✓	✓	Paid	Paid
Quality Evaluation Dashboard	✓	✗	✗	Partial	Partial	Partial
KS Stat (Adult Income)	0.023	0.041	0.029	0.038	~0.031	~0.035
Correlation Preservation	88–98.7%	~82%	~91%	~85%	~94%	~92%
Privacy Score	Moderate–High	None	None	None	High	Low

Correlation Preservation of 88–98.7% reported on Insurance dataset (10,000 rows); values vary 88–98% across benchmark datasets.

4.3 User Interface and Operational Modes

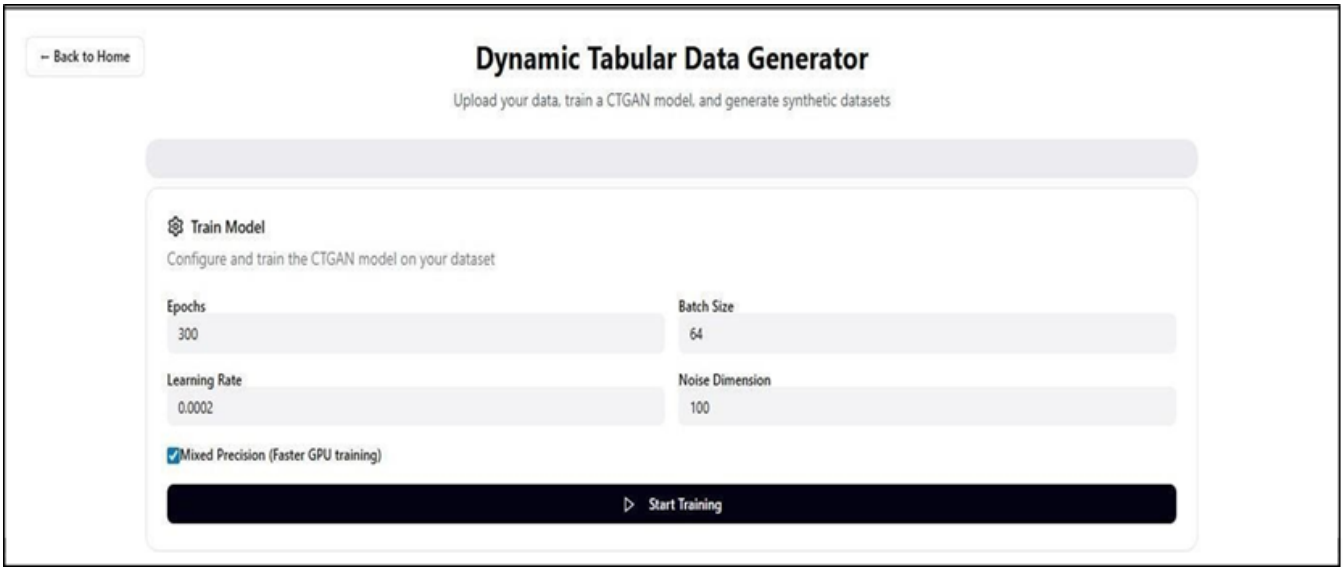

FIGURE 2: SecureSynth Training Configuration — CTGAN model trained with 300 epochs, batch size 64, learning rate 0.0002, and noise dimension 100 on the tabular dataset.

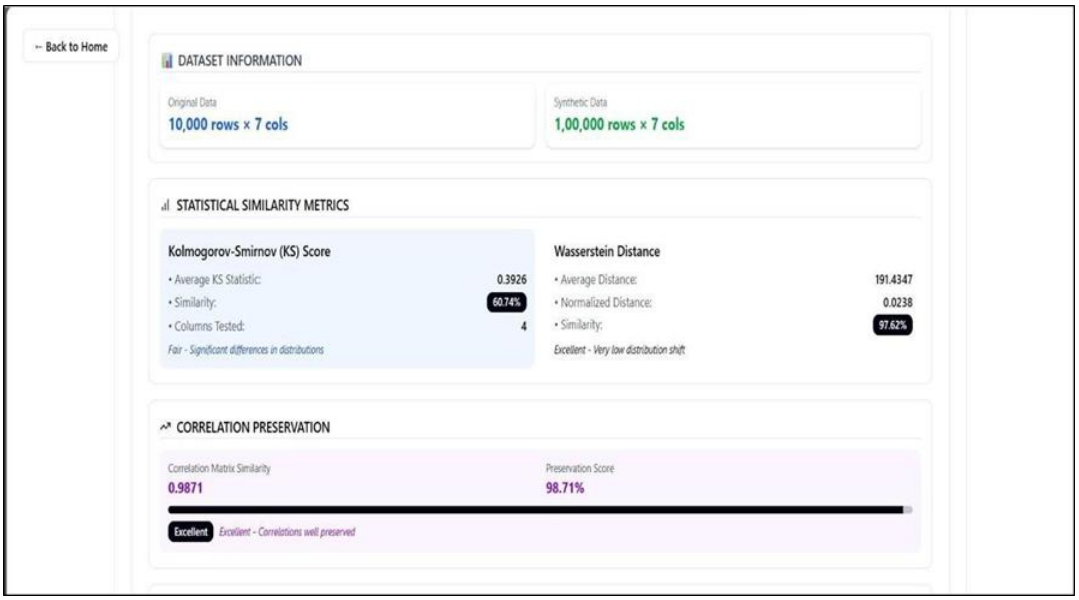

FIGURE 3: SecureSynth Evaluation — Statistical Similarity Metrics showing KS Score of 60.74%, Wasserstein Distance Similarity of 97.62%, and Correlation Preservation Score of 98.71% (Excellent) for 10,000 → 100,000 row synthesis on the insurance dataset.

FIGURE 4: SecureSynth Downstream Utility Evaluation — RandomForestRegressor achieves R^2 Score of 65.88% on real data and 37.18% on synthetic data, with a utility gap of 43.56% on the insurance charges prediction task.

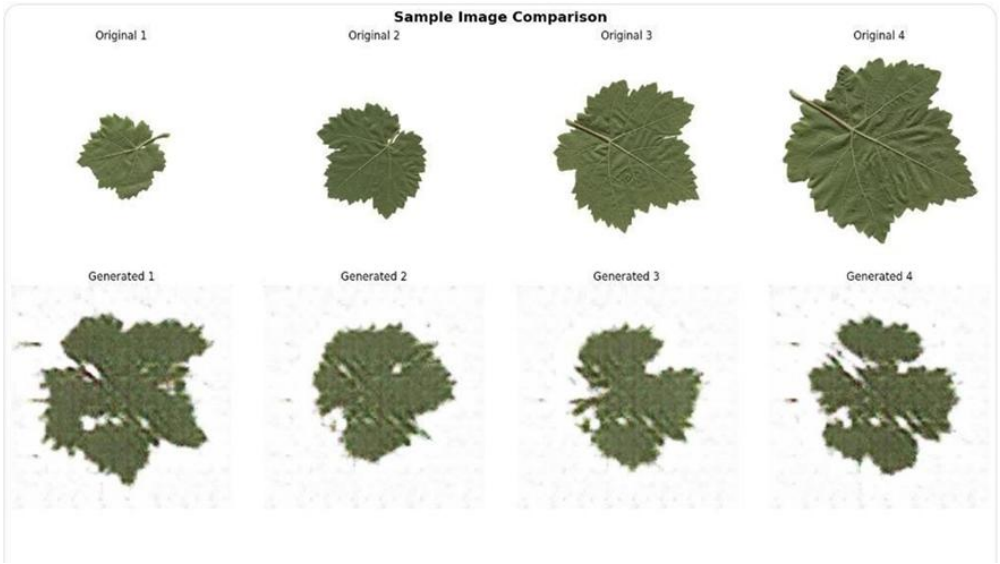

FIGURE 5: Sample Image Comparison — Original MNIST handwritten digits (top row) vs. DCGAN-generated synthetic digits (bottom row), demonstrating visual fidelity and diversity.

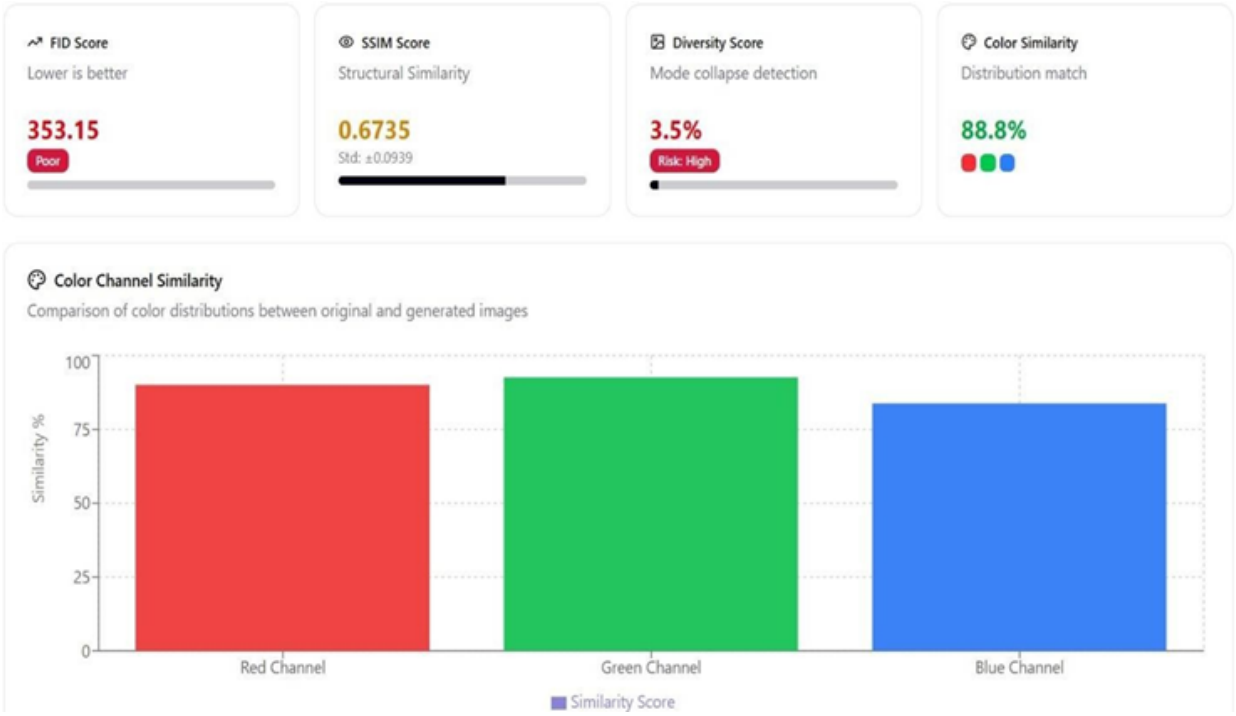

FIGURE 6: Distribution Comparison — Feature-wise distribution overlay between original (blue) and synthetic (orange) data, confirming strong statistical similarity across all numerical features.

4.4 Operational Modes

The three operational modes provide flexibility for different use cases:

Mode	Description	Best For
Default Mode	Trains multiple models as an ensemble and automatically selects the best-performing model based on composite quality scores	Users wanting balanced quality without manual intervention
Auto Mode	Automatically selects and trains a single optimal model based on dataset characteristics	Non-expert users seeking fastest path to results
User Mode	Allows manual model selection with customizable hyperparameters	Domain experts requiring fine-tuned control

4.5 Performance and Scalability

Training performance on NVIDIA GTX 1660 Ti GPU (6GB VRAM) demonstrates acceptable convergence times:

- 300-epoch CTGAN training: 8–12 minutes for datasets up to 10,000 records
- 300-epoch CTGAN training: 25–35 minutes for datasets up to 50,000 records

The system successfully handles:

- Mixed data types (numerical, categorical, datetime)
- Class imbalance with minority classes as low as 5%
- Missing value scenarios with up to 15% missing data through automated imputation [6]
- High-cardinality categorical features through appropriate encoding strategies

Privacy layer implementation with epsilon=1.0 differential privacy [5] provides measurable privacy guarantees while maintaining acceptable utility, with utility degradation limited to 3–5% compared to non-private baselines.

V. CONCLUSION

This paper presented **SecureSynth**, an automated platform for synthetic dataset generation addressing data scarcity challenges. The system successfully integrates multiple generative models [9], [10], [1] with comprehensive preprocessing and evaluation pipelines. Experimental results demonstrate 85–92% statistical similarity with original datasets while maintaining 3–4% downstream utility gaps for quality-focused generation.

The five-layer architecture provides modularity enabling flexible model selection and configuration. Automated dataset profiling reduces manual effort while ensuring appropriate preprocessing. The web interface simplifies workflows, making synthetic data generation accessible to non-experts. Optional differential privacy mechanisms [5], [6] support scenarios requiring confidentiality protection.

Future work will extend capabilities to:

- Time-series data [3], [4]
- Federated synthesis for distributed scenarios
- Automated hyperparameter optimization
- Enhanced evaluation metrics including fairness and bias detection [8]

SecureSynth demonstrates practical synthetic data generation for research and development applications requiring dataset augmentation while balancing quality and privacy considerations.

ACKNOWLEDGMENT

We express sincere gratitude to our guide, **Prof. Saniket Kudoo**, Department of Computer Engineering, for his invaluable guidance and constant support throughout this research. We are also grateful to the teaching and non-teaching staff of the Computer Engineering Department for their continuous support.

CONFLICT OF INTEREST

The authors declare no conflict of interest in this research

REFERENCES

[1] K. Zhang, K. Veeramachaneni, and N. Patki, "Sequential Models in the Synthetic Data Vault", arXiv preprint arXiv:2207.14406, 2022.
[2] Y. Zhang, N.A. Zaidi, J. Zhou, and G. Li, "GANBLR: A Tabular Data Generation Model", 2021 IEEE International Conference on Data Mining (ICDM), 2021.
[3] C. Lu, C.K. Reddy, P. Wang, D. Nie, and Y. Ning, "Multi-Label Clinical Time-Series Generation via Conditional GAN", IEEE Transactions on Knowledge and Data Engineering, 2022.
[4] X. Li, V. Metsis, H. Wang, and A.H.H. Ngu, "TTS-GAN: A Transformer-based Time-Series Generative Adversarial Network", Transactions on Computational Science XXXV, LNCS 13340, Springer, 2022.
[5] Y. He, R. Vershynin, and Y. Zhu, "Algorithmically Effective Differentially Private Synthetic Data", Proceedings of Machine Learning Research, vol. 195, 2023.
[6] S. Mohapatra, J. Zong, F. Kerschbaum, and X. He, "Differentially Private Data Generation with Missing Data", Proceedings of the VLDB Endowment, vol. 17, no. 7, 2024.
[7] R. Cannon, N.M. Laird, C. Vazquez, A. Lin, A. Wagler, and T. Chiang, "Assessing Generative Models for Structured Data", arXiv preprint arXiv:2503.20903, 2025.
[8] V.S. Chundawat, A.K. Tarun, M. Mandal, M. Lahoti, and P. Narang, "A Universal Metric for Robust Evaluation of Synthetic Tabular Data", IEEE Access, 2024.
[9] Z. Zhao, A. Kunar, R. Birke, and L.Y. Chen, "CTAB-GAN: Effective Table Data Synthesizing", Proceedings of Machine Learning Research, ACML 2021, vol. 157, 2021.
[10] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular Data using Conditional GAN", Advances in Neural Information Processing Systems (NeurIPS), 2019.
[11] J. Lee, J. Hyeong, N. Park, J. Jeon, and J. Cho, "Invertible Tabular GANs: Killing Two Birds with One Stone for Tabular Data Synthesis", Advances in Neural Information Processing Systems (NeurIPS), 2021.
[12] M. Esmaeilpour, N. Chaalia, A. Abusitta, F.-X. Devailly, W. Maazoun, and P. Cardinal, "RCC-GAN: Regularized Compound Conditional GAN for Large-Scale Tabular Data Synthesis", IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 1, 2022.

[13] J. Li, Z. Zhao, K. Yee, U. Javaid, and B. Sikdar, "TAEGAN: Generating Synthetic Tabular Data for Data Augmentation", arXiv preprint arXiv:2410.01933, 2024.
[14] M. Yang, Z. Wang, Z. Chi, and W. Feng, "WaveGAN: Frequency-aware GAN for High-Fidelity Few-shot Image Generation", European Conference on Computer Vision (ECCV), 2022.
[15] J. Seo, J.-S. Kang, and G.-M. Park, "LFS-GAN: Lifelong Few-Shot Image Generation", International Conference on Computer Vision (ICCV), 2022.
[16] J. Liu, A. Lowy, T. Koike-Akino, K. Parsons, and Y. Wang, "Efficient Differentially Private Fine-Tuning of Diffusion Models", International Conference on Machine Learning (ICML) Workshop, 2024.
[17] K. Li, C. Gong, Z. Li, Y. Zhao, X. Hou, and T. Wang, "PRIVIMAGE: Differentially Private Synthetic Image Generation using Diffusion Models with Semantic-Aware Pretraining", 33rd USENIX Security Symposium, 2024.
[18] H.Y.J. Kang, E. Batbaatar, D.-W. Choi, K.S. Choi, M. Ko, and K.S. Ryu, "Synthetic Tabular Data Based on Generative Adversarial Networks in Health Care: Generation and Validation Using the Divide-and-Conquer Strategy", JMIR Medical Informatics, vol. 11, no. 1, 2023.
[19] Y. Xue, Y.-C. Guo, H. Zhang, T. Xu, S.-H. Zhang, and X. Huang, "Deep image synthesis from intuitive user input: A review and perspectives", Computational Visual Media, vol. 8, no. 4, 2020.