Generative AI for real-time fraudulent Transaction simulation

Enock Onkarabile Buys

[1] Student Number:219013044
[2] Academy of Computer Science and software Engineering, University of Johannesburg

Abstract.
Financial fraud detection systems are critically inconvenienced by extreme class imbalances, where fraudulent transactions often constitute less than 0.5% of datasets. This imbalance biases machine learning models toward the majority class (legitimate transactions), resulting in poor fraud recall. Traditional oversampling techniques like SMOTE fail to capture the complex, non-linear correlations inherent in financial datasets. This paper proposes and evaluates the application of Generative Adversarial Networks (GANs) to synthesize realistic fraudulent transactions, augmenting the minority class and improving classifier performance. The paper implements a comparative framework across three pipelines: Pipeline 1 employs a custom-coded GAN and Random Forest classifier for foundational control ,Pipeline 2 uses library-based implementations (TensorFlow for GAN and Scikit-learn for Random Forest) for production-level efficiency and Pipeline 3 utilizes CTGAN with XGBoost for tabular data specialization. Experiments assess the impact of synthetic data augmentation at percentages ranging from 5% to 30%, with and without SMOTE preprocessing. Results demonstrate that Pipeline 3 without SMOTE achieves the highest baseline F1-score (0.775) and PR-AUC (0.786), with augmentation providing marginal improvements at lower percentages. SMOTE enhances synthetic quality in Pipelines 2 and 3 but increases training times significantly and often reduces F1-scores in augmented models. The findings validate GAN-based augmentation as an effective method for addressing data scarcity in fraud detection, highlighting trade-offs between custom and library-based approaches, and emphasizing the superiority of no-SMOTE setups for optimal performance.

Keywords: Fraud Detection, Generative Adversarial Networks ,Data Augmentation ,Class Imbalance ,SMOTE ,Pipeline

1 Introduction

The rapid digitization of financial services has precipitated a massive increase in electronic transactions, creating a fertile ground for fraudulent activities. According to the Nilson Report, cumulative global card fraud losses are projected to reach $404 billion over the next decade(Marek, 2025). This escalating threat underscores the urgent need for more effective detection mechanisms. A fundamental challenge in this domain is the extreme class imbalance inherent in transaction datasets, where fraudulent cases often represent less than 1% of all records (Dal Pozzolo et al., 2015). This skew causes machine learning models to become biased toward the majority class (legitimate transactions), severely limiting their ability to identify fraud.

Traditional approaches to mitigate this imbalance include algorithmic techniques like cost-sensitive learning and data-level methods such as the Synthetic Minority Oversampling Technique (SMOTE)(Chawla et al., 2002). However, SMOTE generates samples through linear interpolation, which often fails to produce realistic fraud patterns that preserve the complex, non-linear feature relationships found in real transaction data (Dal Pozzolo et al., 2014). Furthermore, stringent data privacy regulations like the Protection of Personal Information Act (POPIA) in South Africa and the General Data Protection Regulation (GDPR) in Europe limit the sharing of real transaction data, exacerbating the scarcity of fraud examples needed to train robust models (Choudhury and Bhowal, 2015).

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. (2014), offer a promising solution by learning the underlying distribution of real data and generating novel, realistic samples. While GANs have achieved remarkable success in image synthesis, their application to tabular financial data is less explored and presents unique challenges, including training instability, mode collapse, and evaluating generated sample quality (Fiore et al., 2019). Variants like Conditional Tabular GAN (CTGAN) have been developed specifically for tabular data, demonstrating improved fidelity in synthetic generation (Xu et al., 2019).

This paper makes a twofold contribution: Firstly, it presents a practical, implementable solution to data scarcity in fraud detection using standard GANs and specialized variants. Secondly, it provides an empirical comparison between self-coded implementations (demonstrating fundamental understanding) and library-based implementations (demonstrating practical application), offering insights into performance tradeoffs. The research is guided by the question: To what extent can GANs augment imbalanced fraud datasets, and how do custom implementations compare to established libraries in terms of synthetic data quality and final classification performance?

The remainder of the paper is structured as follows: Section 2 reviews the literature on fraud detection and GAN applications. Section 3 describes the proposed model, including the three pipelines. Section 4 details the implementation. Section 5 presents the results. Section 6 provides a critique and analysis. Section 7 concludes the paper and suggests future work.

2 Literature Review

The evolution of fraud detection has progressed from expert-defined rule-based systems to sophisticated machine learning models. Rule-based systems, while interpretable, are static and unable to adapt to novel fraud patterns (Bhattacharyya et al., 2011). Machine learning approaches, including Logistic Regression, Random Forests, and

Support Vector Machines, have improved detection rates by learning patterns from historical data (Adewumi and Akinyelu, 2017). The phenomenon of deep learning has further enhanced capabilities, with models like Long Short-Term Memory (LSTM) networks proving effective in capturing sequential fraud patterns (Jurgovsky et al., 2018).

Despite these advances, the class imbalance problem persists. Dal Pozzolo et al. (2015) extensively documented this issue, showing that models trained on imbalanced data suffer from low recall. Techniques like random undersampling discard potentially useful majority class data, while SMOTE can generate noisy samples that degrade classifier performance (Bounab et al., 2024). Hybrid methods combining SMOTE with Edited Nearest Neighbors (ENN) have been proposed to address noise, demonstrating improved performance in Medicare fraud detection (Bounab et al., 2024, Singh et al., 2021).

GANs have emerged as a powerful tool for generating synthetic data in imbalanced scenarios. Fiore et al. (2019) applied GANs to credit card fraud detection, showing that augmented datasets improve minority class recall. For tabular data, CTGAN conditions the generation process on class labels, enabling better capture of multivariate distributions (Xu et al., 2019). Recent applications include using tabular GANs for high-cardinality categorical data in fraud contexts and synthetic transactional datasets for payment fraud detection (Ramachandran et al., 2023).

Ensemble models like Random Forest and XGBoost have been widely used in fraud detection due to their robustness. Akazue et al. (2023) enhanced Random Forest with ensemble feature selection for fraud detection, achieving higher accuracy. For XGBoost, Patel et al. (2025) optimized it with hybrid SMOTE-ENN for real-time credit card fraud detection, addressing class imbalance effectively. Priscilla and Prabha (2020) investigated optimized XGBoost for handling class imbalance in credit card fraud.

This review highlights the potential of GANs in fraud detection but identifies gaps in comparative evaluations between custom and library-based implementations, which this study addresses.

3 Methodology

This research proposes a comprehensive framework leveraging Generative Adversarial Networks (GANs) to synthesize realistic fraudulent transactions for addressing class imbalance in financial fraud detection. The methodology employs three distinct pipelines representing different implementation paradigms: from foundational custom coding to production-ready libraries and specialized tabular data approaches. This multi-pipeline design enables rigorous comparison of synthetic data quality, classification performance, and implementation trade-offs.

3.1 Data Preprocessing and Experimental Conditions

All implementations had consistent experimental circumstances thanks to a uniform preprocessing pipeline.

Dataset Characteristics: The study used a dataset of 284,807 financial transactions that included 30 features, such as transaction time, amount, and anonymized attributes (V1-V28) that were obtained by PCA transformation. With a fraud incidence of roughly 0.17%, the binary target variable Class separates valid (0) transactions from fraudulent (1) transactions. Data Partitioning: Stratified sampling maintained the original class distribution by dividing the dataset into training (188,678 transactions), validation (53,908 transactions), and testing (26,954 transactions) groups. For upcoming predictive modeling, an extra 5% reserve was kept. Experimental Conditions: To assess the interaction effects between traditional and generative oversampling, extensive testing was conducted across two preprocessing paradigms: 10,000 fraud samples with SMOTE oversampling and 313 original fraud data without SMOTE.

3.2 Pipeline Architectures

This pipeline embodies a from-scratch implementation designed to demonstrate a fundamental understanding of algorithmic and architectural principles in generative modeling. The Generator architecture consists of a fully connected neural network that processes 64-dimensional latent vectors through hidden layers with 128 and 256 neurons, respectively. Each layer employs the LeakyReLU activation function with an α parameter of 0.2 and includes batch normalization to stabilize learning and accelerate convergence. The output layer produces a 30-dimensional vector with a tanh activation function to align with the normalized input data range. The Discriminator architecture mirrors the generator's structure, accepting 30-dimensional input vectors and passing them through hidden layers of 256 and 128 neurons. It similarly employs the LeakyReLU activation function and culminates in a sigmoid output layer for binary real/fake classification.

The Classifier component is a custom Random Forest algorithm comprising 150 decision trees, each with a maximum depth of 10 and a minimum sample split of 10, using the Gini impurity criterion for node splitting. The Training methodology follows a classical adversarial learning approach utilizing binary cross-entropy loss and custom optimization with differentiated learning rates of 0.0001 for the generator and 0.0004 for the discriminator, ensuring stable and balanced training dynamics.

The second pipeline leverages established libraries to enhance stability and production readiness. It implements a Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) using TensorFlow and follows a mirrored architecture similar to Pipeline 1. A gradient penalty coefficient ($\lambda = 10$) is applied to enforce the Lipschitz constraint, improving training stability and mitigating mode collapse. The Classifier for this pipeline utilizes Scikit-learn's RandomForestClassifier with the same hyperparameters as in Pipeline 1 (n_estimators = 150, max_depth = 10, min_samples_split = 10) to ensure a controlled comparative analysis. The Training methodology employs Keras's built-in train_on_batch function under the WGAN-GP framework, optimized with a learning rate of 0.00005. This configuration allows for stable gradient updates and improved convergence consistency.

The third pipeline integrates state-of-the-art generative modeling techniques tailored for tabular data synthesis. The Generator architecture is based on the Conditional Tabular GAN (CTGAN), which applies mode-specific normalization for continuous variables and conditional training based on fraud labels to enhance representational learning. It employs a training-by-sampling mechanism to effectively address distributional imbalances within the dataset. The Classifier component utilizes the XGBoost algorithm, configured with optimized hyperparameters for imbalanced classification tasks and early stopping based on validation performance to prevent overfitting.

3.3 Data Augmentation Strategy

For each pipeline, GANs were trained exclusively on minority class samples under both SMOTE conditions. Post-training, generators synthesized fraudulent transactions at precisely controlled percentages of the original fraud count: 5%, 10%, 12%, 15%, 20%, and 30%. Augmented datasets were constructed by appending synthetic samples to original training data, while maintaining pristine validation and test sets for unbiased evaluation.

3.4 Evaluation Framework

he evaluation framework for the proposed pipelines encompasses both classification performance and synthetic data quality. The primary classification metrics include the F1-Score which represents the harmonic mean of precision and recall and the Precision-Recall Area Under the Curve (PR-AUC), which provides a robust measure of performance under class imbalance conditions, particularly relevant in fraud detection contexts.

The secondary metrics comprise the Receiver Operating Characteristic Area Under the Curve (ROC-AUC), overall classification accuracy, and training computational efficiency. These additional measures offer a broader perspective on model reliability, discrimination capability, and practical usability in real-world deployment scenarios.

For synthetic quality assessment, the evaluation focuses on statistical fidelity between real and generated data. This is quantified through Mean Difference (Mean Diff), Standard Deviation Difference (Std Diff), and Kullback–Leibler Divergence (KL Div), each capturing distinct aspects of distributional similarity.

4 Implementation

All experiments were conducted on a standardized workstation environment featuring Windows 11 Home Single Language (Version 24H2, OS build 26100.4946) with an AMD Ryzen 5 3550H processor, 16 GB RAM, and NVIDIA GeForce GTX 1650 GPU (4 GB VRAM) to ensure computational consistency and reproducibility. The development environment utilized PyCharm IDE with SQLite3 for database management and Python 3.12.8 as the core programming language. The experimental framework employed a comprehensive dataset comprising 284,807 transactions with 30 features, partitioned through stratified sampling to preserve the inherent fraud ratio of approximately 0.17%. This partitioning resulted in training (188,678 transactions), validation (53,908), and test (26,954) sets, with the remaining 5% reserved for future predictive modeling. Fraud samples for GAN training were maintained at 313 for non-SMOTE conditions and 10,000 for SMOTE-preprocessed scenarios.

The three pipelines were implemented with distinct configurations: Pipeline 1 utilized a custom GAN trained for 200 epochs with batch size 256 and latent dimension 64, paired with a Random Forest classifier employing out-of-bag error estimation; Pipeline 2 leveraged TensorFlow GAN with optimized settings across 200 epochs, integrated with Scikit-learn Random Forest using consistent parameters of 150 trees, maximum depth 10, and minimum sample split of 10; Pipeline 3 implemented CTGAN with 200 epochs and batch size 250, combined with XGBoost classifier featuring early stopping capabilities. Across all pipelines, synthetic data quality was rigorously assessed using Mean Difference (Mean Diff), Standard Deviation Difference (Std Diff), and Kullback-Leibler Divergence (KL Div) metrics, which were aggregated into a composite quality score where higher values indicated superior synthetic fidelity.

The evaluation protocol established comprehensive baselines with non-augmented data (0% synthetic) under both SMOTE conditions, followed by systematic testing across six incremental augmentation levels ranging from 5% to 30% of the original fraud count. Performance validation was conducted exclusively on pristine test sets to prevent data leakage and ensure realistic assessment, while computational efficiency was monitored through precise recording of training times for both GAN synthesis and classifier training phases across all experimental conditions.

5 Results

GAN training results and synthetic quality are summarized in Table 1.

Pipeline	SMOTE	Final G Loss	Final D Loss	Synthetic Quality	Training Time (s)	Fraud Samples
1	No	-995.32	1099.52	0.533	35	313

1	Yes	-27066.12	25572.72	0.504	603	10,000
2	No	0.70	-0.23	0.912	22	313
2	Yes	0.75	-0.25	0.945	66	10,000
3	No	-2.99	0.30	0.314	47	313
3	Yes	-0.60	-0.90	0.566	924	10,000

Table 1: GAN Training and Synthetic Quality

Baseline and augmented performance are shown in Table 2 (averages across augmentation levels for brevity).

Pipeline	**Model**	**SMOTE**	**Synthetic %**	**Avg F1**	**Avg ROC-AUC**	**Avg PR-AUC**
1	RF	No	0% (Baseline)	0.609	0.883	0.681
1	RF	No	5-30%	0.597	0.867	0.661
1	RF	Yes	0% (Baseline)	0.649	0.983	0.627
1	RF	Yes	5-30%	0.655	0.972	0.648
2	RF	No	0% (Baseline)	0.769	0.965	0.737
2	RF	No	5-30%	0.762	0.968	0.727
2	RF	Yes	0% (Baseline)	0.554	0.977	0.688
2	RF	Yes	5-30%	0.478	0.987	0.684

3	XGBoost	No	0% (Baseline)	0.775	0.986	0.786
3	XGBoost	No	5-30%	0.745	0.979	0.742
3	XGBoost	Yes	0% (Baseline)	0.590	0.982	0.744
3	XGBoost	Yes	5-30%	0.517	0.986	0.728

Table 2: Model Performance (Test Metrics)

Accuracy remained high (>0.997), but F1 and PR-AUC highlight minority class performance.

6 Critique and Analysis

The results reveal key insights into GAN-based augmentation for fraud detection. Pipeline 2 consistently produced the highest synthetic quality (0.912–0.945), attributed to its stable WGAN architecture, outperforming the custom Pipeline 1, which exhibited loss explosion and lower quality (0.504–0.533). Pipeline 3 (CTGAN) showed moderate quality, improving with SMOTE from 0.314 to 0.566, aligning with findings in tabular data generation (Ramachandran et al., 2023, Xu et al., 2019).

Without SMOTE, baselines were stronger, with Pipeline 3 achieving the top F1 (0.775) and PR-AUC (0.786), surpassing Random Forest baselines in Pipelines 1 and 2. This supports XGBoost's efficacy in imbalanced fraud scenarios (Patel et al., 2025). Augmentation provided mixed benefits: In no-SMOTE setups, it slightly improved Pipeline 1 (peak F1 0.667 at 20%) but hovered around baselines for Pipelines 2 and 3, suggesting high-quality synthetics add value only up to 15–20% before introducing noise.

SMOTE boosted quality in Pipelines 2 and 3 but inflated training times (10–20x) and often degraded F1 in augmented models, possibly due to overfitting on oversampled data. For instance, Pipeline 2's augmented F1 dropped to 0.478 average with SMOTE, versus 0.762 without. This indicates SMOTE is beneficial for GAN stability but counterproductive for final performance in balanced training.

Limitations include GAN instability in Pipeline 1 and lack of real-time deployment testing. t-SNE visualization, as suggested, could confirm synthetic-real alignment but was not implemented here thus future work should incorporate it. Overall, no-SMOTE with low-percentage augmentation (5–15%) optimizes efficiency and performance, validating GANs as superior to SMOTE alone for complex fraud patterns.

7 Conclusion

This study demonstrates that GAN-based synthetic augmentation effectively addresses class imbalance in financial fraud detection, with empirical evidence supporting its superiority over traditional methods like SMOTE in capturing non-linear data distributions. The comparative framework across three pipelines reveals that library-based approaches (Pipelines 2 and 3) outperform custom implementations in synthetic quality and efficiency, achieving up to 0.945 quality scores and training times as low as 22 seconds without SMOTE. Pipeline 3's XGBoost baseline without SMOTE emerged as the optimal configuration, yielding the highest F1-score (0.775) and PR-AUC (0.786), underscoring XGBoost's robustness for imbalanced tabular data.

Augmentation at 5–20% levels provides marginal but consistent improvements in no-SMOTE scenarios, peaking at F1 0.769 in Pipeline 2, while higher percentages risk performance degradation due to noise. SMOTE, while enhancing GAN quality, significantly increases computational overhead and often reduces downstream F1-scores, suggesting its selective use only when fraud samples are critically scarce. These findings align with prior research (Bounab et al., 2024, Xu et al., 2019) and contribute a validated methodology for practical fraud systems, emphasizing trade-offs in custom vs. production-ready tools.

Future research should explore hybrid SMOTE-GAN variants, real-time integration, and t-SNE-based quality assessments to further refine synthetic fidelity. Ultimately, this work advances generative AI's role in simulating realistic fraudulent transactions, paving the way for more resilient financial security systems amid rising global fraud threats projected at $404 billion over the next decade (Marek, 2025).

References

ADEWUMI, A. O. & AKINYELU, A. A. 2017. A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *International Journal of System Assurance Engineering and Management,* 8**,** 937–953.

AKAZUE, M. I., DEBEKEME, I. A., EDJE, A. E., ASUAI, C. & OSAME, U. J. 2023. Unmasking fraudsters: ensemble features selection to enhance random forest fraud detection. *Journal of Computing Theories and Applications,* 1**,** 201–211.

BHATTACHARYYA, S., JHA, S., THARAKUNNEL, K. & WESTLAND, J. C. 2011. Data mining for credit card fraud: A comparative study. *Decision support systems,* 50**,** 602–613.

BOUNAB, R., ZAROUR, K., GUELIB, B. & KHLIFA, N. 2024. Enhancing medicare fraud detection through machine learning: Addressing class imbalance with SMOTE-ENN. *IEEE Access,* 12**,** 54382–54396.

CHAWLA, N. V., BOWYER, K. W., HALL, L. O. & KEGELMEYER, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research,* 16**,** 321–357.

CHOUDHURY, S. & BHOWAL, A. Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection. 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015. IEEE, 89–95.

DAL POZZOLO, A., CAELEN, O., JOHNSON, R. A. & BONTEMPI, G. Calibrating probability with undersampling for unbalanced classification. 2015 IEEE symposium series on computational intelligence, 2015. IEEE, 159–166.

DAL POZZOLO, A., CAELEN, O., LE BORGNE, Y.-A., WATERSCHOOT, S. & BONTEMPI, G. 2014. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications,* 41**,** 4915–4928.

FIORE, U., DE SANTIS, A., PERLA, F., ZANETTI, P. & PALMIERI, F. 2019. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences,* 479**,** 448–455.

GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. & BENGIO, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems,* 27.

JURGOVSKY, J., GRANITZER, M., ZIEGLER, K., CALABRETTO, S., PORTIER, P.-E., HE-GUELTON, L. & CAELEN, O. 2018. Sequence classification for credit-card fraud detection. *Expert systems with applications,* 100**,** 234–245.

MAREK, L. 2025. *Card fraud losses will increase over next decade* [Online]. Available: https://www.paymentsdive.com/news/payments-fraud-losses-prevention-nilson-outlook/737440 [Accessed October 13 2025].

PATEL, M. H., PATEL, M. & SAVANI, M. K. 2025. An Optimized XGBoost Framework for Real-Time Credit Card Fraud Detection: Addressing Class Imbalance with Hybrid SMOTE-ENN Resampling. *International Journal of Scientific Research in Science and Technology,* 12**,** 1129–1136.

PRISCILLA, C. V. & PRABHA, D. P. Influence of optimizing XGBoost to handle class imbalance in credit card fraud detection. 2020 third international conference on smart systems and inventive technology (ICSSIT), 2020. IEEE, 1309–1315.

RAMACHANDRAN, K., KAYATHWAL, K., WADHWA, H. & DHAMA, G. FraudAmmo: Large scale synthetic transactional dataset for payment fraud detection. 2023 International Joint Conference on Neural Networks (IJCNN), 2023. IEEE, 1–7.

SINGH, S., KAYATHWAL, K., WADHWA, H. & DHAMA, G. Metgan: Memory efficient tabular gan for high cardinality categorical datasets. International Conference on Neural Information Processing, 2021. Springer, 519–527.

XU, L., SKOULARIDOU, M., CUESTA-INFANTE, A. & VEERAMACHANENI, K. 2019. Modeling tabular data using conditional gan. *Advances in neural information processing systems,* 32.