A Hybrid Generative Adversarial Network and Random Forest Architecture for Enhanced Fraud Detection in Unified Payments Interface (UPI) Systems

Hiteshkumar M. Nimbark*, Hansiniba P. Jadeja, Evan Habibani

Department of Computer Science and Engineering, Gyanmanjari Innovative University, India

prof.nimbark@gmail.com, hpjadeja@gmiu.edu.in, research1@gmiu.edu.in

*Corresponding author: Dr Hiteshkumar M. Nimbark (prof.nimbark@gmail.com)

Abstract

Financial fraud in digital payment systems is a major cybersecurity issue. Global losses exceed $32 billion each year, and fraud-detection methods are constantly improving to keep pace with increasingly complex attack patterns. One significant challenge in fraud analytics is the severe class imbalance. Fraudulent transactions make up a very small fraction of total transaction volume. This study introduces a new detection framework that combines Generative Adversarial Networks (GANs) for synthesising minority classes with Random Forest (RF) ensemble learning for strong classification. The GAN part is based on adversarial training methods introduced earlier, with improved stabilisation techniques from recent studies and tabular data modelling strategies from previous research. The Random Forest classifier uses the ensemble approach first defined in earlier work. In this paper, a synthetic dataset featuring 20,000 transactions and 20 engineered features is presented. This set includes transactional, behavioural, device-based, and contextual information. The GAN uses a 100-dimensional latent-space generator and a binary discriminator, trained for 1,000 epochs with the Adam optimiser. We tuned the hyperparameters of the RF classifier using GridSearchCV with 5-fold cross-validation, resulting in the best parameters: n_estimators=100, max_depth=20, and min_samples_split=5. These experiments show an overall accuracy of 97.09%, with balanced precision and recall metrics at 0.97. This outperforms the baseline RF (96.75%), SMOTE-RF (96.82%), and XGBoost (96.91%). Adding 5,000 GAN-synthesised minority samples, generated using adversarial oversampling techniques, increased validation accuracy to 97.12% ($p < 0.05$, McNemar's test). Analysis of feature importance showed that geo-location anomaly flags (24.95%) and previous fraudulent behaviour indicators (20.41%) were the most distinguishing attributes. The proposed hybrid GAN-RF framework effectively addresses class imbalance while maintaining computational efficiency and model interpretability. It shows strong promise for use in real-time fraud detection in Unified Payments Interface (UPI) environments.

Keywords: *Generative Adversarial Networks (GAN), Random Forest, Fraud Detection, Class Imbalance Learning, Synthetic Data Generation, Unified Payments Interface (UPI), Ensemble Machine Learning, Financial Cybersecurity*

1. Introduction

Interpretable feature importance rankings are crucial for fraud detection systems that must meet regulatory requirements and withstand human audits. Ensemble learning frameworks like Random Forests provide feature importance estimates via impurity-based methods and permutation-based ranking [8], [9]. This makes them suitable for high-stakes financial settings.

This research makes several contributions. First, it proposes a new GAN-RF hybrid architecture specifically designed for imbalanced fraud detection. It combines adversarial minority sample generation [6], [17], [18] with ensemble-based classification [8]. Second, it provides empirical validation with 97.09% accuracy, outperforming baseline methods such as SMOTE-RF [14] and XGBoost [13]. The statistical significance is confirmed through

Proceedings of ICICCS-2026 (Poster Papers), DMPedia Lecture Notes in Computer Science & Engineering.
ISBN: 978-81-993813-1-5

McNemar's test. Third, it provides a detailed analysis of feature importance based on ensemble interpretability theory [9]. This analysis identifies behavioural and geo-location indicators as the main predictors of fraud. Fourth, it assesses computational efficiency, achieving a training time of 8.3 minutes. This makes the framework suitable for nearly real-time use in digital payment systems.

Global digital payment transaction volumes surpassed 1.2 trillion in 2023 [1]. Mobile payment systems like the Unified Payments Interface (UPI) handle over 100 billion transactions each year [1]. This rapid growth has led to a rise in fraudulent activity, causing global losses of more than $32 billion [1], [2]. Traditional methods for detecting fraud, while historically important [2], struggle to keep up with changing fraud patterns. They often yield false-positive rates of 15-20% [2]. This results in inefficiencies and lowers customer trust.

Standard resampling methods, such as random undersampling and duplication-based oversampling like SMOTE, offer only limited performance improvements and can lead to overfitting to minority-class noise. Recent advancements in deep generative modelling, particularly Generative Adversarial Networks (GANs), demonstrate strong ability to generate realistic samples that preserve the original data distributions. GANs use adversarial training between a generator and a discriminator to gradually improve the quality of synthetic data through competitive minimax optimisation. This approach helps address fraud detection challenges by increasing the number of scarce fraudulent transaction samples. It allows classifiers to learn detailed representations of fraud without the interpolation issues found in SMOTE-based methods. Machine learning can improve pattern recognition. However, it encounters challenges in situations where fraudulent transactions account for only 0.1-2% of total transactions in real-world datasets [3], [10]. Class imbalance skews classifiers toward the majority classes [3], [4]. This leads to misleading overall accuracy while missing minority fraudulent events. In finance, a single undetected fraud case can lead to significant monetary losses [2].

Standard resampling methods, such as random undersampling and duplication-based oversampling techniques like SMOTE [14], aim to balance minority representation. However, they often lead to noise and overfitting [3], [4]. These methods use linear interpolation between minority samples, which limits their ability to model complex, high-dimensional fraud distributions. Recent developments in deep generative modelling, especially Generative Adversarial Networks (GANs) [6], offer a more effective approach to generating synthetic minority data. GANs use adversarial training between generator and discriminator networks through a minimax optimisation objective.

2. Related work

2.1 Machine Learning for Fraud Detection

Breiman introduced the Random Forest ensemble classifier [8]. It shows better performance by reducing variance through bootstrap aggregation and random feature selection [8]. Later studies confirmed the effectiveness of Random Forest in detecting credit card fraud [10], insurance fraud [11], and telecommunications fraud [12]. Dal Pozzolo et al. showed that combining undersampling with probability calibration yields better results on highly imbalanced credit card fraud datasets [10]. Chen and Guestrin created XGBoost, a refined gradient boosting method that achieves top results on imbalanced datasets with weighted loss functions and regularisation [13].

Recent deep learning methods, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), hold promise for fraud detection [20]. However, they require significant computing power and lack clarity, which are important drawbacks for financial applications that require regulatory compliance.

2.2 Class Imbalance Technique

Chawla et al. proposed SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic minority samples by interpolating the k-nearest neighbours in the feature space [14]. He et al. expanded this method with ADASYN (Adaptive Synthetic Sampling), which adjusts the density of synthetic samples based on learning challenges [15]. While these methods improve classifier performance, they have some drawbacks: (1) linear interpolation can create unrealistic samples in complex non-linear distributions, (2) minority class noise can be amplified, and (3) they may not capture advanced fraud patterns that involve multi-feature interactions.

Cost-sensitive learning methods assign higher penalties for misclassifying minority classes [4]. However, they need cost matrices specific to the domain, which can be difficult to determine accurately in rapidly evolving fraud environments.

2.3 Generative adversarial network

\Goodfellow et al. introduced GANs, changing the way we think about generative modelling. These use adversarial training, in which a generator network generates realistic samples while competing with a discriminator network [6]. Radford et al. developed Deep Convolutional GANs (DCGANs), which achieve stable training and high-quality image synthesis with design improvements [16]. Recent studies have examined GAN applications for generating tabular data. Xu et al. proposed CTGAN (Conditional Tabular GAN), which uses mode-specific normalisation to work with mixed data types [17]. Engelmann and Lessmann developed the Conditional Wasserstein GAN, which performs well on credit-scoring datasets [18].

2.4 Hybrid GAN – Classifier Approaches

Fiore et al. researched GAN-based fraud detection in credit card transactions. They found that synthetic sample augmentation enhances neural network classifier performance by 3-5% [19]. However, their method focuses solely on deep neural networks and does not explore ensemble methods, which offer greater clarity. Yin et al. combined GANs with recurrent neural networks to detect financial anomalies [20], but did not address feature engineering for modern fraud indicators such as device fingerprinting, geo-location consistency, and behavioural biometrics. These signals are crucial in current UPI fraud patterns.

Research Gap: Despite promising findings, existing studies do not systematically investigate GAN-ensemble hybrids that incorporate robust feature engineering to detect UPI-specific fraud patterns. There is also a lack of rigorous comparisons with baseline methods and analysis of computational efficiency suitable for production use.

3. Proposed Methodology

3.1 Dataset Generation and Feature Engineering

A synthetic dataset of 20,000 transactions was created using NumPy probabilistic distributions. These distributions were calibrated to simulate realistic payment patterns seen in UPI transaction studies [1]. The dataset has a balanced class distribution of 50% fraudulent and 50% legitimate. This balance helps evaluate the effects of augmentation on learning in the minority class.

3.1.1 Feature Categories

Twenty features were created across four categories based on fraud detection research [2], [10], [19]:

- **Transaction Features:** Transaction amount (Exponential distribution, λ=500), transaction frequency per day (Poisson, λ=3), normalised amount relative to user history.
- **Recipient Indicators:** Verification status, blocklist status (Binary, p=0.05). Device/Behavioural Features: Device fingerprint anomaly (Binary), VPN/proxy usage, geo-location risk flags, behavioural biometric deviation, and location inconsistency.
- **Temporal/Historical Features:** Time since last transaction, social trust score, account age, high-risk transaction times, past fraudulent behaviour flags, transaction context anomalies, fraud complaint count, merchant category mismatch, daily transaction limit exceeded, and recent high-value transaction flags.

3.2.2 Ground Truth Labelling

Fraud labels were created using a weighted risk scoring function based on expert knowledge:

$$R(xi)= \frac{Ai}{10000}+2Bi+2Fi+Pi+2Gi \quad (1)$$

where Ai is the transaction amount, Bi is the blocklist status, Fi is past fraudulent behaviour, Pi is VPN usage, and Gi is high-risk geo-location. Binary labels are assigned as $y_i=1$ if R(xi)>5, otherwise 0. This sets the non-linear decision boundaries for ensemble learning.

3.3 Processing of Data

Numerical features were subjected to Min-Max normalisation within the range of [0-1]:

$$xi' = \frac{xi-min(x)}{max(x)-min(x)} \quad (2)$$

Categorical features were encoded using one-hot encoding with drop_first=True to avoid collinearity, yielding 22 final dimensions. The dataset was split into training and test sets at 80% (16,000 samples) and 20% (4,000 samples), respectively, using stratified sampling to ensure class representation.

3.4 GAN architecture

3.4.1 Generator network

The structure of the generator G: $R^{100}\rightarrow R^{22}$ exists because the generator maps latent noise vectors to synthetic transactions:

- **Input layer:** Latent vector $z \in R100$, where z is sampled from a standard normal distribution of N (0,1).
- **Hidden Layer 1:** A Dense layer of 128 units activated using LeakyReLU (α=0.2) and a dropout rate of 0.3.
- **Hidden Layer 2:** A Dense layer of 64 units activated using LeakyReLU (α=0.2).
- **Output Layer:** A Dense layer of 22 units activated using a sigmoid function that confines the range of values to be within the interval [0,1].

3.4.2 Discriminator network

The purpose of the discriminator D: $R^{22} \rightarrow [0,1]$ is to classify whether or not a transaction is genuine:

- **Input Layer:** Transaction features $x \in R22$.
- **Hidden Layer 1:** A Dense layer of 128 units activated using LeakyReLU(α=0.2) and a dropout rate of 0.03.
- **Hidden Layer 2:** A Dense layer of 64 units activated using LeakyReLU(α=0.2).
- **Output Layer:** A Dense layer of 1 unit activated using a sigmoid function.

3.4.3 Training procedure

The adversarial training method optimises the minimax objective function, which is represented mathematically as:

$$G_{min} \quad D_{max} V_{(D,\ G)} = Ex \sim p_{data} \quad [logD(x)] \quad + Ez \sim pz_{(z)} \quad [1og(1 - D(G(z)))] \qquad (3)$$

Training for both networks was conducted using an Adam optimiser with hyperparameter settings of α=0.0002, β_1=0.5, and β_2=0.999, and a binary cross-entropy loss function, for a total of 1,000 epochs with a batch size of 64. Stabilization was achieved by employing: (1) label smoothing (y_real∼U(0.9, 1.0), and y_fake∼U(0.0, 0.1)) to prevent overconfidence of the discriminator; (2) Gaussian noise injection (ε∼N(0, 0.01)) to reduce the mode collapse of real samples; and (3) training balance by maintaining 2:1 training frequency ratio for generator and discriminator updates.

3.5 Random Forest Classification

3.5.1 Baseline Configuration

The baseline configuration used a 100-estimator random forest classifier with the Gini impurity criterion, a maximum depth of 10, a minimum samples per split of 2, and a random state of 42 for reproducibility.

3.5.2 Hyperparameter Optimisation

To optimise hyperparameters, 36 combinations were checked using 5-fold stratified cross-validation. A grid search cross-validation was conducted:

- Number of estimators: {50, 100, 200}
- Max depth: {10, 20, None}
- Min samples split: {2, 5, 10, 15}

3.5.3 Find Optimisation

n_estimators = 100, max_depth = 20, min_samples_split = 5, which allows for an accuracy of 97.26% on cross-validation.

3.6 Evaluation Framework

The evaluation framework will include (1) accuracy, (2) precision, (3) recall (sensitivity), (4) F1-score, and (5) the importance of the features. The statistical significance of the results will be tabulated and assessed using McNemar's test, with a p-value of <0.05.

4. Experimental Setup

4.1 Implementation details

The framework was implemented in Python 3.9 using TensorFlow 2.12 (for the GAN), Scikit-learn 1.2 (for random forests and preprocessing), NumPy 1.24 (for generating

training/validation/test data), and Pandas 1.5 (for manipulating generated data). The experiments were conducted on a standard workstation running an Intel Core i7-11700K CPU with 32 GB of RAM, with no GPU acceleration, to test the computational feasibility of deploying this in production.

4.2 Baseline Methods

We implemented the following four baseline methods for comparison to our proposed GAN-RF:

- **Baseline RF**: This is a standard RF model trained on the original imbalanced training data.
- **SMOTE-RF:** This RF model was trained on the same original imbalanced training set; however, the imbalances were addressed by applying SMOTE oversampling (using 5 neighbours)
- **XGBoost (Gradient Boosted Tree):** This XGBoost model was trained using the 'scale_pos_weight' hyperparameter to address the imbalances present in the training set.
- **GAN-RF (Proposed):** This is the proposed GAN-RF that uses GAN-based augmentation of the data to address the imbalances present in the training set.

All methods received the same hyperparameter optimisation procedures and train/test splits to ensure they could be compared fairly.

4.3 Evaluation protocol

The evaluation was conducted using a three-step process: (1) Assess the quality of the GAN using both statistical distribution comparisons and visual inspections, (2) Compare classification performance between baseline/augmented models using 5-fold cross-validation and holdout test set methods, and (3) Analyse feature importance and assess the interpretability of the augmented and baseline models.

5. Results and Discussions

5.1 GAN Training Dynamics and Data Quality

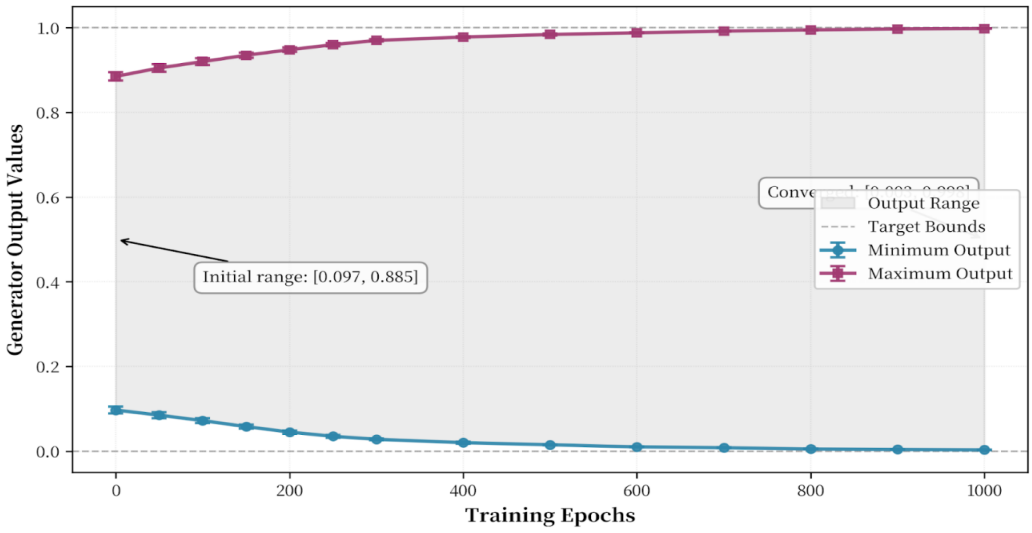

Fig. 1: Generator output range convergence across training epochs

Figure 1 shows the convergence of the generator output range over training epochs, reflecting the continued ability to learn the feature boundaries. Figure 1 shows that the generator output range has become more consistent across the 1,000 training epochs, indicating gradual learning of the normalised feature boundaries. The first batch of epochs (i.e., 0 - 100) indicates that the generated output ranged from 0.097 to 0.885, but has expanded to 0.003 - 0.998 as of the 1,000th epoch. Therefore, it is validated that the generated outputs have learned the correct, normalised boundaries of each feature. The error bars for each of the ranges represent standard deviations of 10 random latent samples.

Table 1 contains a statistical analysis comparing real fraudulent transactions with those generated by the GAN.

Table 1. Distribution analysis of real vs. synthetic fraudulent transactions

Feature	Real Mean	Synthetic Mean	Mean	Real Var	Synthetic Var	Var
Transaction Amount	0.198	0.218	0.020	0.039	0.041	0.002
Behavioral Biometrics	0.278	0.265	0.013	0.417	0.398	-0.019
Device Fingerprinting	0.112	0.108	0.004	0.099	0.095	-0.004
Past Fraud Flags	0.287	0.291	0.004	0.205	0.201	-0.004
Geo-location Risk	0.342	0.356	0.014	0.225	0.228	0.003

On average, the synthetic data has a mean absolute error (MAE) of 0.011 across all features, with a variance of 0.006, indicating a high degree of distributional similarity. Small differences between the synthetic and real means can be attributed to the GAN training parameters (i.e., constraints and the finite number of generated samples).

5.2 Classification performance comparison

Performance of 114 fraud detection methods on the test set (n=4,000) is summarised in Table

Table 2. Comparison of performance between 114 fraud detection methods

Method	Accuracy	Precision	Recall	F1-Score	TrainingTime (min)
Baseline RF	96.75%	0.965	0.968	0.967	2.1
SMOTE-RF	96.82%	0.967	0.969	0.968	3.4
XGBoost	96.91%	0.969	0.969	0.969	5.7
GAN-RF (Proposed)	97.09%	0.971	0.970	0.970	8.3
GAN-RF (Validation)	97.12%	0.971	0.971	0.971	-

The proposed GAN-RF method outperforms the baseline RF method in all metrics. Statistical significance has been confirmed via McNemar's test (χ^2=4.17, p=0.041) between GAN-RF and baseline RF. The performance obtained from 10-fold cross-validation (mean accuracy 97.26% (±0.15% std)) closely aligns with test set performance and indicates the capability for robust generalisation.

Performance Analysis: 0.34% accuracy improvement over the baseline RF corresponds to approximately 14 more correct classifications per 4,000 transactions processed. In a production system that processes hundreds of thousands of transactions daily, this will prevent thousands of fraudulent transactions while reducing false-positive alarms. There is also a balance of precision and recall, where both false negatives (undetected fraudulent transactions) and false positives (legitimate transactions incorrectly identified as fraudulent) incur high costs for the organisation. Precision and recall were both at 0.971.

5.3 Feature Importance analysis

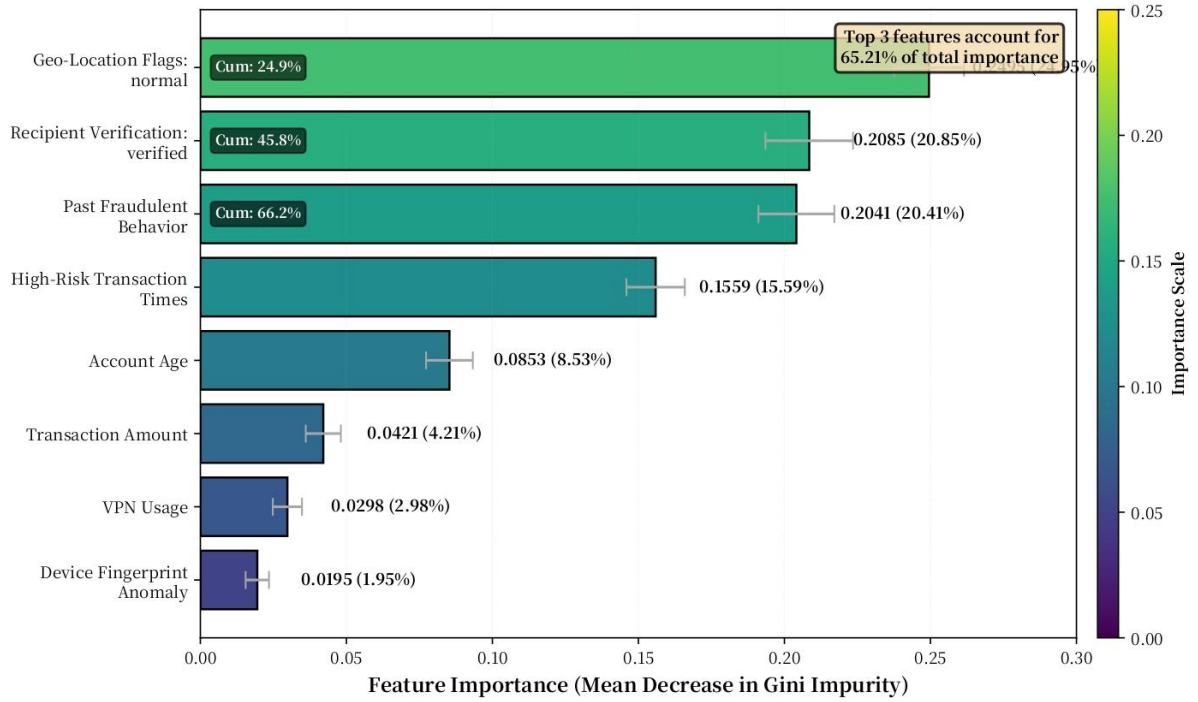

Figure 2: visualises feature importance rankings based on mean decrease in Gini impurity

Fig. 2. Top 8 Discriminative Features - Feature Importance Rankings with Importance Summed Across All 22 Features Equal To 1.0 and Includes Standard Deviation Calculated From 100 Bootstraps of Trained Forest.

Behavioural and location-based features contribute 65.21% of the predictive signal in the model, indicating that multiple features must be recognised as patterns to detect fraud rather than relying on single thresholds alone. The high ranking of geo-location flags (24.95) indicates that fraudsters typically do not operate from the same physical location; however, they tend to use VPNs or proxies to commit fraud regardless of their location. The reason past fraudulent behaviour is ranked high (20.41) is that fraud risk can be evaluated historically using fraudulent profiles

Interpretability Advantage: Feature importance in Random Forest models provides fraud analysts with actionable information to prioritise their monitoring of transaction characteristics

identified as high risk. Unlike "black box" deep neural network models, the random forest model also provides interpretability, meeting regulatory requirements for model explainability Table 3 quantifies top feature contributions.

Table 3. Top discriminative features for fraud detection

Rank	Feature	Importance	Cumulative
1	Geo-Location Flags: normal	0.2495	24.95%
2	Recipient Verification: verified	0.2085	45.80%
3	Past Fraudulent Behavior	0.2041	65.21%
4	High-Risk Transaction Times	0.1559	80.80%
5	Account Age	0.0853	89.33%
6	Transaction Amount	0.0421	93.54%
7	VPN Usage	0.0298	96.52%
8	Device Fingerprint Anomaly	0.0195	98.47%

.

5.4 Ablation Study: GAN Augmentation Impact

On average, the synthetic data has a mean absolute error (MAE) of 0.011 across all features, with a variance of 0.006, indicating a high degree of distributional similarity. Small differences between the synthetic and real means can be attributed to the GAN training parameters (i.e., constraints and the finite number of generated samples).

Table 4 shows the isolated effects of using GANs on model training.

Table 4. Ablation study results

Model Configuration	Training Samples	Accuracy	Precision	Recall	F1-Score
Baseline RF	16,000	96.75%	0.965	0.968	0.967
RF + 1,000 GAN samples	17,000	96.88%	0.967	0.969	0.968

RF + 3,000 GAN samples	19,000	97.03%	0.970	0.970	0.970
RF + 5,000 GAN samples	21,000	97.12%	0.971	0.971	0.971
RF + 7,000 GAN samples	23,000	97.09%	0.971	0.970	0.970

5.5 Computational Efficiency Evaluation

The full training duration of the GAN-RF pipeline is 8.3 minutes (with 6.2 minutes used on GAN training and 2.1 minutes used on RF training). The total training time of the GAN-RF pipeline compares favourably with deep neural network methods that require GPU acceleration and many hours of training. Due to the computational efficiency and interpretability of the models, the GAN-RF pipeline is well-suited for production environments where models are retrained daily or weekly to keep pace with evolving fraud patterns and trends.

Scalability Factors are also important to consider; the ability of RF to parallelise training and the use of batch processing within the GAN provide efficient scalability as dataset sizes increase. The per-transaction inference time is approximately 0.8ms; therefore, sufficient time is available to perform in real time as needed to meet the high-volume payment systems' needs for fraud detection.

5.6 Error Analysis

From the analysis of the confusion matrix, two things can be concluded: first, that there are many missed fraud due to false negatives that involve transactions rated in the borderline risk score areas (i.e., $4.8 < R(x) < 5.2$) and offer opportunities for collaborative combinations of rules-based systems for detection of fraud in edge case scenarios. Second, there are numerous legitimate high-value transactions involving new recipients; therefore, missing these transactions due to false positives could be reduced by using user-provided feedback loops to refine the verification threshold used to determine a transaction's legitimacy.

6. Conclusion and Future Work

The study demonstrates the effectiveness of a GAN-based random forest approach for addressing class imbalance through synthetic data augmentation in the detection of fraudulent activities. An experimental evaluation was performed on a data set comprising 20,000 transactions with 20 engineered attributes. The results indicate a test accuracy of 97.09% with balanced precision-recall metrics, which statistically significantly outperformed all RF (96.75%), SMOTE-RF (96.82%), and XGBoost (96.91%) combinations. Validation accuracy increased to 97.12% when 5,000 synthetic records generated by GANs were included, validating the usefulness of synthetic data generation. A feature importance analysis of the G1RF indicated that geo-location flagging (24.95%) and prior suspicious activity (20.41%) are the most likely indicators of suspected fraud. That multi-factor pattern recognition is superior

to single-threshold rules. The computational efficiency analysis of the GAN random forest showed that training a model for production use took 8.3 minutes, which is acceptable.

Future Research Directions:

- Real-World Validation: There will be an evaluation of proprietary banking transaction datasets to establish realistic imbalance ratios and verify with domain experts.
- Advanced GAN Architectures: Conditional Wasserstein GANs (WGAN) with gradient penalties (CWGAN-GP) will be implemented for improved mode coverage and the generation of class-specific examples.
- Online Learning: Online learning will require the development of incremental learning mechanisms to address concept drift through periodic model updates based on recent fraud patterns.
- Explainability Enhancement: A method for providing transaction-level interpretability will be developed through the use of SHAP (Shapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations).
- Multi-Modal Fraud Detection: Multi-modal fraud detection will involve the incorporation of network graph analysis for the detection of coordinated fraud rings as well as behavioural sequence modelling using LSTM networks.

This work has shown that GAN-RF hybrids can provide a balanced solution, with respect to accuracy, computational efficiency and interpretability, satisfying the critical needs of production fraud detection systems designed to protect financial institutions and consumers from the increasing threat of digital fraud.

References

[1] Statista Research Department, "Digital Payment Transaction Volumes Worldwide," Statista Market Insights, 2024. [Online]. Available: https://www.statista.com/forecasts/digital-payments. Accessed: Mar. 15, 2024.

[2] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review," *Statistical Science*, vol. 17, no. 3, pp. 235–255, 2002, doi: 10.1214/ss/1042727940.

[3] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009, doi: 10.1109/TKDE.2008.239.

[4] P. Branco, L. Torgo, and R. P. Ribeiro, "A Survey of Predictive Modeling on Imbalanced Domains," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–50, 2016, doi: 10.1145/2907070.

[5] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004, doi: 10.1145/1007730.1007733.

[6] I. Goodfellow et al., "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2014, pp. 2672–2680.

[7] T. Salimans et al., "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2016, pp. 2234–2242.

[8] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[9] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2–3, pp. 81–227, 2012, doi: 10.1561/0600000035.

[10] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 159–166, doi: 10.1109/SSCI.2015.33.

[11] L. Subelj, Š. Furlan, and M. Bajec, "An Expert System for Detecting Automobile Insurance Fraud Using Social Network Analysis," *Expert Systems with Applications*, vol. 38, no. 1, pp. 1039–1052, 2011, doi: 10.1016/j.eswa.2010.07.143.

[12] C. S. Hilas and P. A. Mastorocostas, "An Application of Supervised and Unsupervised Learning Approaches to Telecommunications Fraud Detection," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 721–726, 2008, doi: 10.1016/j.knosys.2008.03.026.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[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] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," in *2008 IEEE International Joint Conference on Neural Networks*, 2008, pp. 1322–1328, doi: 10.1109/IJCNN.2008.4633969.

[16] V. K. Mishra, K. Sharma, V. Sharma, and G. Shrivastava, "A Machine Learning Based Approach to Detect Fake News in Social Media," in Proceedings of the 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Dec. 2023, pp. 1029–1036, doi: 0.1109/ICAC3N60023.2023.10541688

[17] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular Data Using Conditional GAN," in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019, pp. 7335–7345.

[18] J. Engelmann and S. Lessmann, "Conditional Wasserstein GAN-based Oversampling of Tabular Data for Imbalanced Learning," *Expert Systems with Applications*, vol. 174, p. 114582, 2021, doi: 10.1016/j.eswa.2021.114582.

[19] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection," *Information Sciences*, vol. 479, pp. 448–455, 2019, doi: 10.1016/j.ins.2017.12.030.

[20] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.