Financial Fraud Detection Using Ensemble Learning and Resampling Techniques

Ehsan Bagheri, Zachary Moran, Quinton Wilson

1 Introduction

Financial fraud presents an ongoing and increasingly complex threat to global economic stability, individual financial security, and the integrity of financial institutions. As digital transactions continue to proliferate, so do the opportunities and complexities associated with fraudulent activities. Traditional fraud detection methods, including rule-based systems, have provided foundational security measures. However, these methods frequently fall short, as they lack adaptability and responsiveness to the rapidly evolving and sophisticated nature of modern fraud schemes, often leveraging advanced artificial intelligence and machine learning technologies.

Addressing these limitations requires sophisticated machine learning (ML) methodologies capable of dynamically adapting to emerging fraud patterns. The significant imbalance characteristic of fraud datasets, where fraudulent transactions typically comprise a very small fraction of total transactions, exacerbates the difficulty of accurate detection. In this study, we utilize a comprehensive dataset derived from financial transactions initially obtained from Kaggle. Due to limitations identified in the initial dataset, particularly regarding the extreme class imbalance and inadequate detail, we generated a customized synthetic dataset specifically tailored to our research requirements. Our project aims to optimize fraud detection performance, achieving high recall rates to minimize false negatives (missed frauds) and low false-positive rates to ensure practical usability. We propose a robust methodological framework that includes data preprocessing, targeted feature engineering, and advanced ensemble learning techniques to tackle these critical challenges effectively. We compared the results of these different methods ultimately to answer how different ensemble methods and different tuning methodologies compare when addressing class imbalance in fraud detection.

2 Related Work

The problem of fraud detection has garnered considerable attention within the machine learning research community. Historically, financial institutions have relied heavily on rule-based detection systems. While effective for simple fraud patterns, these systems lack flexibility, struggling to adapt to novel and increasingly complex fraud scenarios (Bolton & Hand, 2002). Consequently, supervised machine learning methods have gained prominence, primarily due to their ability to model complex patterns and interactions in large, diverse datasets. Logistic regression, decision trees, support vector machines, and various ensemble learning methods have seen extensive application in fraud detection due to their predictive performance (Phua et al., 2010).

However, despite their strengths, supervised learning methods often encounter significant challenges stemming from class imbalance—a condition wherein fraudulent transactions constitute a minuscule fraction of total transactions. This imbalance frequently biases models towards the majority class, reducing detection rates for fraud cases significantly. Various resampling techniques, such as Synthetic Minority Over-sampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN), have been introduced to address these imbalances effectively. SMOTE, for instance, synthetically generates new minority-class examples, significantly improving model recall rates and overall performance (Chawla et al., 2002).

Ensemble methods have increasingly emerged as powerful solutions to imbalanced classification problems. Random Forest and gradient boosting algorithms such as XGBoost and LightGBM have demonstrated substantial predictive improvements, attributable to their capacity to capture complex, non-linear interactions within high-dimensional data spaces (Dal Pozzolo et al., 2015; Chen & Guestrin, 2016). Recent literature also emphasizes the necessity of addressing dynamic fraud patterns, suggesting the integration of online learning frameworks or temporal dynamics to enhance model adaptability and accuracy (Carcillo et al., 2018). Liu et al. (2020) systematically compared multiple SMOTE variants, providing insights into their effectiveness across various fraud detection tasks.

Despite the significant predictive performance improvements achieved, a major ongoing concern relates to model interpretability and transparency. Many advanced ML techniques, including ensemble and deep learning models, often operate as "black boxes," obscuring internal decision-making processes. This lack of interpretability poses significant regulatory compliance challenges, prompting recent research to explore Explainable Artificial Intelligence (XAI) methodologies. These approaches aim to balance predictive accuracy and interpretability, ensuring models remain both powerful and transparent for stakeholders (Arya et al., 2019).

3 Methods

3.1 Data Creation and Preprocessing

Initially, we utilized a publicly available transactional dataset from Kaggle. However, this dataset posed challenges for our modeling objectives, primarily due to extreme class imbalance and limited transactional detail. Consequently, we developed our own synthetic dataset, designed explicitly for our project needs, featuring over 100,000 transactions with an intentionally maintained fraud rate of approximately 2%.

Our synthetic dataset incorporates twelve comprehensive features designed to capture realistic transactional patterns. Core transactional variables include unique transaction IDs, timestamps, and transaction amounts generated using merchant-specific statistical distributions. The dataset encompasses nine distinct merchant categories: Grocery, Gas, Restaurant, Online, ATM, Retail, Pharmacy, Entertainment, and Unknown, each with tailored transaction amount distributions—for instance, grocery transactions follow a gamma distribution (shape=2, scale=25) plus a $10 base amount, while ATM withdrawals are restricted to common denominations ($20, $40, $60, $80, $100, $200).

Account-level features include randomly generated account numbers, dynamically calculated account creation dates (ensuring accounts predate their transactions), and credit scores drawn from normal distributions with different parameters for legitimate (mean=650, std=100) versus fraudulent accounts (mean=580, std=120). Behavioral variables capture transaction frequency patterns through exponentially distributed previous transaction hours, number of daily transactions, and average monthly spending profiles that differ systematically between fraudulent and legitimate patterns.

To reflect realistic fraud scenarios, location-based features were probabilistically engineered across eight global cities, with legitimate transactions heavily concentrated in Chapel Hill, USA (75% probability) while fraudulent transactions exhibit more dispersed international patterns, particularly favoring London, UK (25%) and Sydney, Australia (20%). Merchant categories were also weighted differently between transaction types, with fraudulent transactions showing elevated probabilities for inherently riskier categories like online shopping (35% vs. standard weighting) and entertainment industries, while maintaining lower probabilities for routine categories like grocery and pharmacy purchases. This probabilistic approach ensures that our synthetic data captures realistic fraud indicators while maintaining sufficient complexity for robust model training and evaluation.

Data preprocessing involved addressing missing data predominantly through median imputation methods, ensuring data integrity and consistency. Numeric features underwent robust scaling to effectively mitigate the disproportionate influence of potential outliers. A hybrid encoding strategy was implemented for categorical features: one-hot encoding was applied to features with low cardinality (transaction types and devices), while high-cardinality features (merchant categories and locations) employed target encoding to efficiently reduce dimensionality.

Our exploratory data analysis of the synthetic dataset revealed several key insights. The correlation heatmap, shown in Figure A3, provides a comprehensive overview of the relationships between our numeric features. A strong positive correlation is indicated by values close to 1, while a strong negative correlation is represented by values near -1. As shown in the heatmap, num_transactions_today exhibits a moderate positive correlation with the is_fraud label (r≈0.43), suggesting that a higher number of transactions in a single day may be indicative of fraudulent activity. Conversely, is_fraud shows weak or no correlation with other features like amount, prev_transaction_hours, avg_monthly_spending, and credit_score, with correlation coefficients close to zero.

The bar plot, shown in Figure A4, illustrates the average transaction amount across different merchant_type categories, segmented by whether the transaction was fraudulent (is_fraud = 1) or not (is_fraud = 0). The plot reveals that fraudulent transactions generally have a higher average amount than non-fraudulent ones across most merchant categories. This difference is particularly pronounced in categories such as ATM, Retail, and Unknown, where the mean amount for fraudulent transactions is notably higher. This finding aligns with the hypothesis that fraudulent transactions often involve larger sums of money. The plot also highlights that certain merchant types, like Restaurant and Gas, have relatively small differences in average amount between fraudulent and non-fraudulent transactions. The high standard deviation in amounts, represented by the error bars, suggests a wide range of transaction values within each category. This visualization underscores the importance of merchant_type as a potential feature for fraud detection models.

3.2 Modeling and Resampling Techniques

Our modeling approach began with logistic regression, serving as a foundational baseline due to its interpretability and simplicity. Subsequently, we explored more sophisticated ensemble learning methods: Random Forest and XGBoost. Both were selected for their proven capability to handle imbalanced datasets, model complex interactions, and provide critical feature importance insights.

To directly address the pronounced class imbalance inherent to fraud datasets, we examined both random undersampling and oversampling strategies. Random undersampling selectively reduces the majority class samples to balance class distribution, while SMOTE generates synthetic minority class examples to augment minority representation without loss of majority-class information. Systematic evaluations were conducted to compare these resampling techniques against baseline models trained on original, imbalanced data.

3.3 Threshold Tuning

To improve classification performance in the presence of class imbalance, we applied threshold optimization based on Precision-Recall (PR) curve analysis. For each trained model, predicted probabilities on the test set were used to compute precision and recall across a range of thresholds.

The optimal threshold was selected by identifying the point that maximized the F1-score, reflecting the best trade-off between precision and recall for the minority class. This approach was applied consistently across all classifiers—Random Forest, XGBoost, and Neural Network—and across both oversampling (SMOTE) and undersampling strategies.

The tuned thresholds were then used to generate final binary predictions for evaluation, replacing the conventional 0.5 decision boundary.

3.4 Hyperparameter Tuning

To improve model performance and generalization, we applied hyperparameter tuning to both XGBoost and Neural Network classifiers.

For XGBoost, we used GridSearchCV to optimize key parameters such as learning rate, tree depth, number of estimators, subsample ratios, and feature sampling (colsample_bytree). Early stopping and regularization (L1, L2) were incorporated to reduce overfitting, while scale_pos_weight was adjusted based on class imbalance. All configurations were validated using 5-fold stratified cross-validation.

For the Neural Network, we tuned architectural and training hyperparameters, including hidden layer sizes (12, 64, 32), dropout rate (20%), learning rate (0.001), batch size (256), and number of epochs (up to 100). Weighted binary cross-entropy loss and 5-fold cross-validation were used to address class imbalance and ensure robust evaluation. This approach allowed the model to learn nonlinear patterns and serve as a complementary tool to ensemble methods.

3.5 Evaluation Metrics

The evaluation of model performance employed metrics explicitly suited for imbalanced classification tasks, including Precision, Recall, and F1-score.

ROC-AUC and Precision-Recall AUC (PR-AUC) were also utilized as threshold-independent metrics, evaluating the models' overall discriminative capabilities comprehensively. We analyzed confusion matrices to better understand the implications of false positives and negatives, crucial for effective operational decision-making.

4 Results

4.1 Ensemble Models

4.1.1 Performance Metrics

Table 1 summarizes the performance of Random Forest and XGBoost models under SMOTE and undersampling strategies, evaluated using default thresholds. Across all metrics, XGBoost with SMOTE achieved the strongest overall performance, with a precision of 0.8287, recall of 0.8950, and an F1-score of 0.8606. It also recorded the highest accuracy (0.9942) and AUC (0.9981), indicating reliable detection capability for the minority class with minimal false positives.

Random Forest with SMOTE yielded slightly lower but comparable results, achieving an F1-score of 0.8439 and recall of 0.8650. Both models demonstrated strong discrimination as reflected in their near-perfect AUC values. In contrast, the undersampling variants of both models showed a notable shift toward higher recall (0.9900 for Random Forest, 0.9750 for XGBoost) at the expense of precision (0.4962 and 0.4665, respectively), resulting in lower F1-scores.

These findings illustrate the trade-off between precision and recall in imbalanced settings. While undersampling boosted sensitivity to fraudulent cases, it increased false positives, which may be problematic in high-stakes applications. SMOTE provided a more balanced outcome, particularly in preserving precision without significantly sacrificing recall. The consistently high AUC across models further supports the effectiveness of ensemble methods in this domain.

Table 1. Performance of Random Forest (RF) and XGBoost (XGB) with SMOTE and undersampling at default threshold.

Model	Sampling	Threshold	Precision (Class 1)	Recall (Class 1)	F1-score (Class 1)	Accuracy	AUC	Threshold Value
RF	SMOTE	Default	0.8238	0.8650	0.8439	0.9936	0.9976	0.50
RF	Undersample	Default	0.4962	0.9900	0.6611	0.9797	0.9976	0.50
XGB	SMOTE	Default	0.8287	0.8950	0.8606	0.9942	0.9981	0.50
XGB	Undersample	Default	0.4665	0.9750	0.6311	0.9772	0.9978	0.50

4.1.2 Feature Importance

Figure 1 shows the feature importance rankings for the XGBoost model under SMOTE (left) and undersampling (right) strategies. In both cases, *num_transactions_today*, *prev_transaction_hours*, and *location* emerge as the top three predictors, with *num_transactions_today* carrying the most weight—especially under undersampling, where it approaches an importance score of 0.5. This pattern is consistent with the Random Forest model (see Appendix, Fig. A1), indicating that these features are robust indicators of fraudulent behavior regardless of model or sampling approach. Other variables, such as *avg_monthly_spending*, *merchant_type*, and *credit_score*, contributed minimally, reinforcing the stability and generalizability of the top-ranked features.

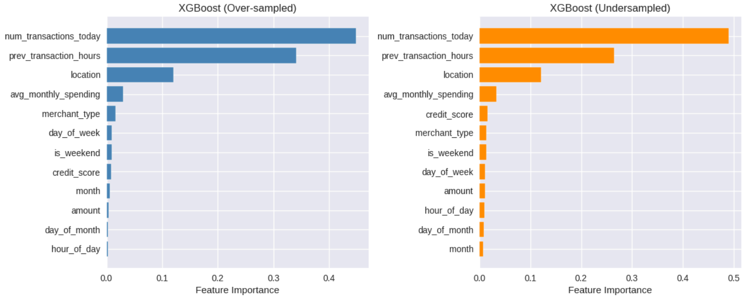


Figure 1. Feature importance from XGBoost models using SMOTE (left) and undersampling (right).

4.2 Neural Network

4.2.1 Performance Metrics

Table 2 presents the classification results for the neural network model trained under SMOTE and undersampling strategies, evaluated at the default threshold of 0.50. The model with SMOTE achieved higher overall performance, particularly in recall (0.9750) and F1-score (0.7847), while maintaining a precision of 0.6566. It also demonstrated high accuracy (0.9893) and the highest AUC across all models tested (0.9986), suggesting excellent separability between fraudulent and non-fraudulent transactions.

In comparison, the undersampled model reached a similar recall (0.9700) but suffered a substantial drop in precision (0.4652), resulting in a lower F1-score (0.6288) and accuracy (0.9771). While the AUC remained strong (0.9968), the imbalance between precision and recall highlights the cost of aggressive undersampling in deep learning settings. Overall, the results suggest that SMOTE is more effective for training neural networks in imbalanced classification tasks, offering a better balance of sensitivity and precision.

Table 2. Performance of neural network (NN) models using SMOTE and undersampling at default threshold (0.50).

Model	Sampling	Threshold	Precision	Recall	F1-score	Accuracy	AUC	Threshold
			(Class 1)	(Class 1)	(Class 1)			Value
NN	SMOTE	Default	0.6566	0.9750	0.7847	0.9893	0.9986	0.50
NN	Undersample	Default	0.4652	0.9700	0.6288	0.9771	0.9968	0.50

4.2.2 Training and Cross-Validation

Figure 2 shows training and validation loss and accuracy curves for neural network models trained with SMOTE (left) and undersampling (right). In both cases, the validation metrics closely track the training metrics, with no signs of divergence. This indicates that the models generalized well and did not suffer from overfitting during training. To further confirm model stability, 5-fold cross-validation was conducted for each sampling strategy. The SMOTE-based model achieved consistently high accuracy (≈0.986), AUC (>0.994), and balanced F1-scores across folds (mean ≈0.73–0.74), with minor variation in precision and recall. The undersampled model showed similarly strong AUC and accuracy metrics, though F1-scores were slightly lower on average. These results reinforce that both training strategies yielded well-generalized models with no evidence of overfitting, and that SMOTE provided better overall balance between precision and recall.

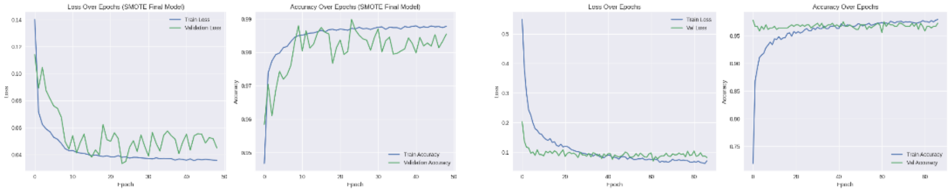


Figure 2. Training and validation loss and accuracy curves for neural networks using SMOTE (left) and undersampling (right).

4.3 Threshold Tuning Impact

Threshold tuning (see Appendix, Fig. A2) significantly improved the performance of all models, particularly for those initially trained with undersampled data (Table 3). As shown in Figure 3, tuning boosted the F1-score of the neural network with undersampling from 0.6288 to 0.8238, and for Random Forest from 0.6611 to 0.8403—a substantial gain in balanced performance. The largest gains were observed where the default threshold (0.50) resulted in high recall but low precision. By adjusting the threshold upward (e.g., 0.94 for NN-Undersample, 0.97 for XGB-Undersample), the models became more conservative in predicting fraud, improving precision and overall F1-score.

Table 3. Performance metrics for all models and sampling strategies after threshold tuning.

Model	Sampling	Threshold	Precision (Class 1)	Recall (Class 1)	F1-score (Class 1)	Accuracy	AUC	Threshold Value
RF	SMOTE	Tuned	0.8194	0.8850	0.8510	0.9938	0.9976	0.46
RF	Undersample	Tuned	0.8261	0.8550	0.8403	0.9935	0.9976	0.88
XGB	SMOTE	Tuned	0.8878	0.8700	0.8788	0.9952	0.9982	0.61
XGB	Undersample	Tuned	0.9086	0.8450	0.8756	0.9952	0.9978	0.97
NN	SMOTE	Tuned	0.8222	0.9250	0.8706	0.9945	0.9986	0.87
NN	Undersample	Tuned	0.8177	0.8300	0.8238	0.9929	0.9968	0.94

Models trained with SMOTE also benefited from tuning, although improvements were more moderate. For instance, the F1-score of the XGBoost model rose from 0.8606 to 0.8788, and the neural network improved from 0.7847 to 0.8706. These enhancements were driven by better precision-recall trade-offs, as seen in Figure 3, which illustrates before-and-after F1-scores. In all cases, tuning shifted thresholds away from 0.50, suggesting that default values are suboptimal in imbalanced classification tasks. Overall, threshold adjustment proved to be a critical step in optimizing model performance, especially for models trained with undersampling.

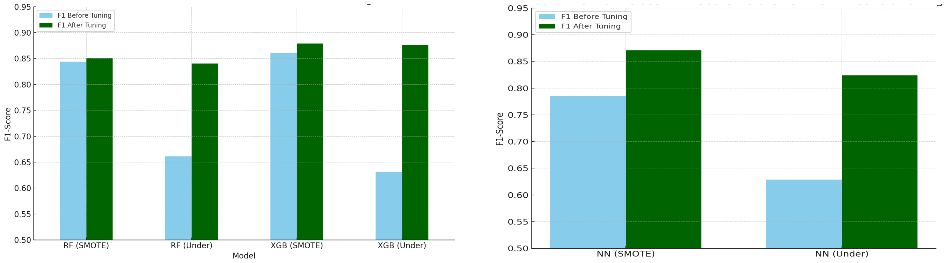


Figure 3. Class 1 F1-Score before vs. after threshold tuning for the left) Random Forest (RF) and XGBoost (XGB), and right) Neural Network (NN)

4.4 Hyperparameter Tuning

Based on the superior performance of SMOTE over undersampling, we focused hyperparameter optimization efforts exclusively on the oversampled data. For XGBoost, we employed GridSearchCV to systematically evaluate combinations of n_estimators (50-300), max_depth (3-9), learning_rate (0.01-0.2), and subsample ratios (0.6-1.0) using 5-fold stratified cross-validation. The optimal configuration (learning_rate=0.2, max_depth=9, n_estimators=300, subsample=0.8) showed negligible performance changes, with test AUC decreasing slightly from 0.9981 to 0.9980 and F1-score remaining unchanged at 0.87. Suggesting that our manual tuning was already optimal.

For the Neural Network, we utilized Optuna's Bayesian optimization to tune architectural parameters including hidden layer sizes, dropout rates (0.1-0.5), learning rates (1e-4 to 1e-2), and batch sizes over 5 trials. The optimization yielded substantial improvements, with cross-validation AUC increasing from 0.9955 to 0.9968, accuracy from 0.9863 to 0.9918, and most importantly, fraud detection F1-score improving from 0.78 to 0.83 on the test set. The tuned Neural Network maintained high recall (0.95) while achieving better precision (0.73 vs 0.66), demonstrating effective balance optimization.

These contrasting results highlight that systematic hyperparameter optimization effectiveness varies significantly across model architectures. While XGBoost showed no meaningful improvement, suggesting our initial manual tuning was already near-optimal for tree-based models, the Neural Network benefited substantially from Bayesian optimization. This finding indicates that neural architectures with complex hyperparameter landscapes require more sophisticated tuning approaches, while simpler models may reach performance plateaus more readily through careful manual parameter selection.

5. Conclusion

In this project, we developed and evaluated fraud detection models using Random Forest, XGBoost, and a neural network across two class imbalance strategies—SMOTE and undersampling—and further enhanced performance through threshold tuning. Our primary accomplishment was demonstrating how model architecture, sampling strategy, and threshold selection jointly influence fraud detection performance, particularly under extreme class imbalance. Among the models evaluated, XGBoost with SMOTE achieved the highest F1-score and AUC at the default threshold, while the neural network with SMOTE recorded the highest AUC overall. Across all settings, the SMOTE-enhanced models consistently outperformed their undersampled counterparts in maintaining precision without sacrificing recall, and our hyperparameter optimization revealed that systematic tuning effectiveness varies significantly across model architectures.

From our analysis, we learned several important insights that extend beyond standard model comparison. First, default decision thresholds (e.g., 0.50) often yield suboptimal trade-offs between precision and recall in imbalanced settings. Our study clearly shows that even high-performing models like XGBoost and neural networks benefit significantly from threshold tuning, which substantially improved F1-scores—especially for undersampled models. Second, the consistency in feature importance rankings across different models and strategies reinforces the reliability of certain behavioral signals, such as *num_transactions_today*, *prev_transaction_hours*, and *location*, as dominant predictors of fraud. Third, our hyperparameter optimization experiments revealed that the effectiveness of systematic tuning varies significantly across model architectures. While XGBoost demonstrated minimal improvement through GridSearchCV (AUC: 0.9981 → 0.9980, F1: 0.87 unchanged), suggesting that careful manual tuning can achieve near-optimal performance for tree-based models, the neural network showed remarkable gains through Bayesian optimization via Optuna, with F1-score improving from 0.78 to 0.83 and precision from 0.66 to 0.73. This finding highlights that neural architectures with complex hyperparameter landscapes benefit substantially from systematic optimization approaches, while simpler models may reach performance plateaus more readily.

Our approach contributes to the field of fraud detection and imbalanced classification by providing a systematic, comparative framework that combines ensemble and neural architectures, interpretable feature analysis, cross-validation, and post-hoc decision threshold optimization. While prior studies often evaluate model performance under a single configuration, our work highlights the compounded effect of preprocessing and decision strategies, presenting a more holistic understanding of the modeling pipeline. By integrating both predictive performance and interpretability, our work helps bridge the gap between model development and practical deployment.

There are several future directions worth exploring. One promising avenue is cost-sensitive learning, where misclassification penalties are explicitly incorporated into the loss function to better reflect the asymmetric costs of false positives and false negatives in fraud scenarios. Additionally, semi-supervised or unsupervised anomaly detection techniques—such as autoencoders, variational models, or isolation forests—could be explored to reduce reliance on labeled data, which is often scarce or noisy in fraud detection. Another direction is real-time or streaming detection, which would require adapting models for online learning, handling concept drift, and maintaining low latency. Further research could also examine model robustness under data shift or adversarial attacks, as fraudsters may intentionally adapt their behaviors to evade detection.

In summary, our work demonstrates that combining model interpretability, sampling strategies, and threshold tuning can yield substantial improvements in fraud detection accuracy. These findings offer practical guidance for deploying models that are both effective and reliable in detecting rare but critical events such as financial fraud.

References

Arya, V., Bellamy, R. K. E., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilovic, A., Mourad, S., Pedemonte, P., Raghavendra, R., Richards, J., Sattigeri, P., Shanmugam, K., Singh, M., Varshney, K. R., Wei, D., & Zhang, Y. (2019). One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. *arXiv preprint arXiv:1909.03012*. https://arxiv.org/abs/1909.03012

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science, 17*(3), 235-255. https://doi.org/10.1214/ss/1042727940

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y. A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). SCARFF: A scalable framework for streaming credit card fraud detection with Spark. *Information Fusion, 41*, 182-194. https://doi.org/10.1016/j.inffus.2017.09.005

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. https://doi.org/10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM. https://doi.org/10.1145/2939672.2939785

Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2015). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications, 41*(10), 4915-4928. https://doi.org/10.1016/j.eswa.2014.02.026

Liu, Z., Cao, W., Gao, Z., Bian, J., Chen, H., Chang, Y., & Liu, T.-Y. (2020). Self-paced ensemble for highly imbalanced massive data classification. *IEEE Transactions on Knowledge and Data Engineering, 32*(6), 1146-1160. https://arxiv.org/abs/1909.03500

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2853. http://jmlr.org/papers/v12/pedregosa11a.html

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*. https://arxiv.org/abs/1009.6119

Appendix

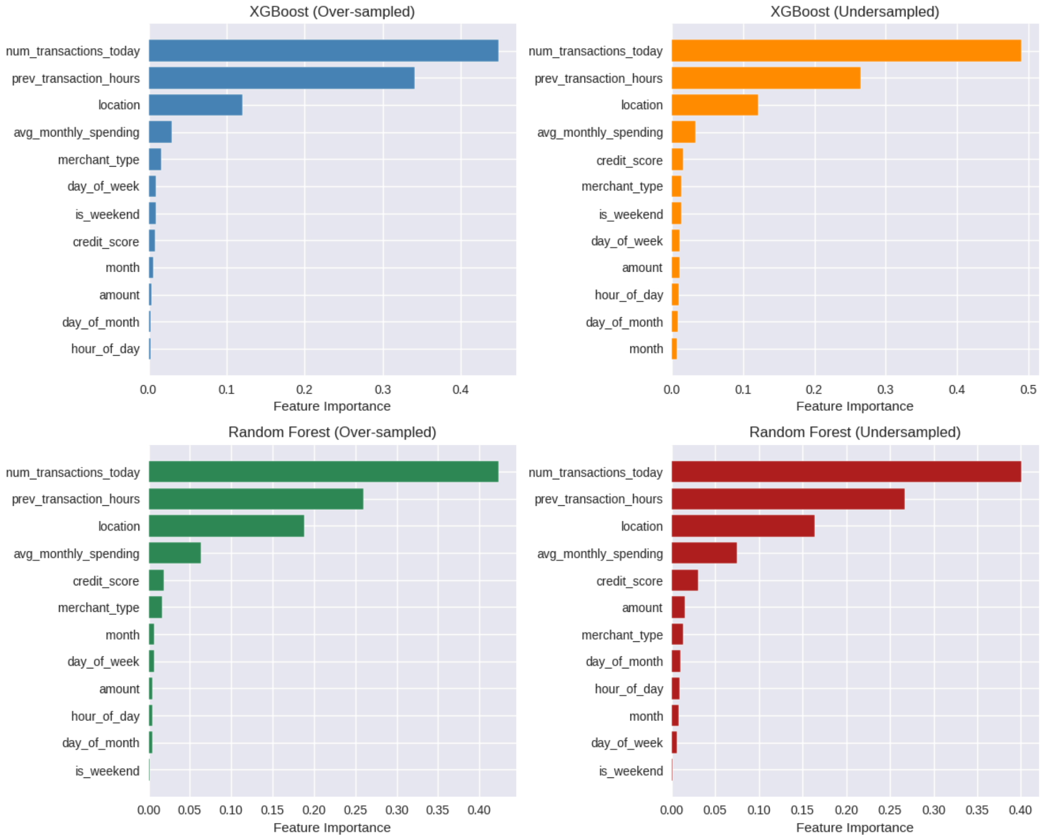


Figure A1. Feature importance rankings for XGBoost (top row) and Random Forest (bottom row) under SMOTE (left) and undersampling (right) strategies.

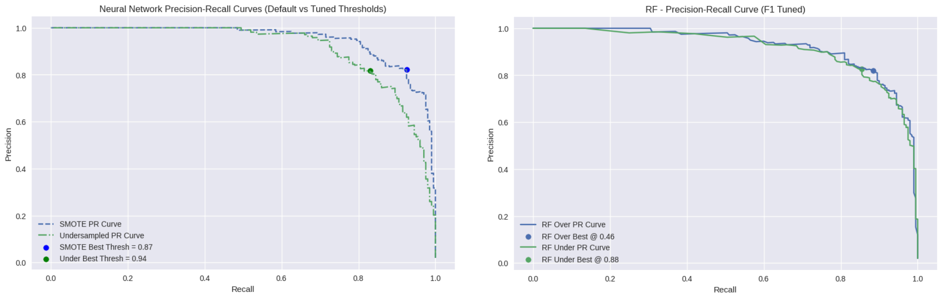
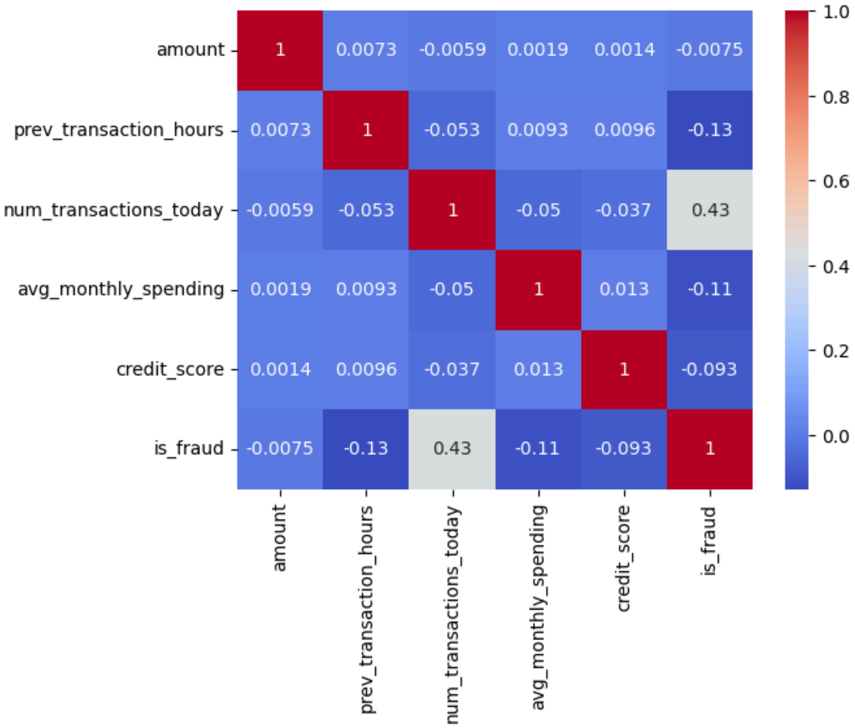


Figure A2. Precision-Recall curves for Neural Network (left) and Random Forest (right) models comparing SMOTE and undersampling strategies under default and tuned thresholds. Points indicate the best F1-score thresholds for each model.



	amount	prev_transaction_hours	num_transactions_today	avg_monthly_spending	credit_score	is_fraud
amount	1	0.0073	-0.0059	0.0019	0.0014	-0.0075
prev_transaction_hours	0.0073	1	-0.053	0.0093	0.0096	-0.13
num_transactions_today	-0.0059	-0.053	1	-0.05	-0.037	0.43
avg_monthly_spending	0.0019	0.0093	-0.05	1	0.013	-0.11
credit_score	0.0014	0.0096	-0.037	0.013	1	-0.093
is_fraud	-0.0075	-0.13	0.43	-0.11	-0.093	1

Figure A3. A correlation heatmap visualizing the relationships between various numeric features in the dataset. The color intensity and numerical values indicate the strength and direction of the correlation, with a value of 1 representing a perfect positive correlation, -1 representing a perfect negative correlation, and 0 representing no correlation. The plot highlights the moderate positive correlation between num_transactions_today and the is_fraud label.

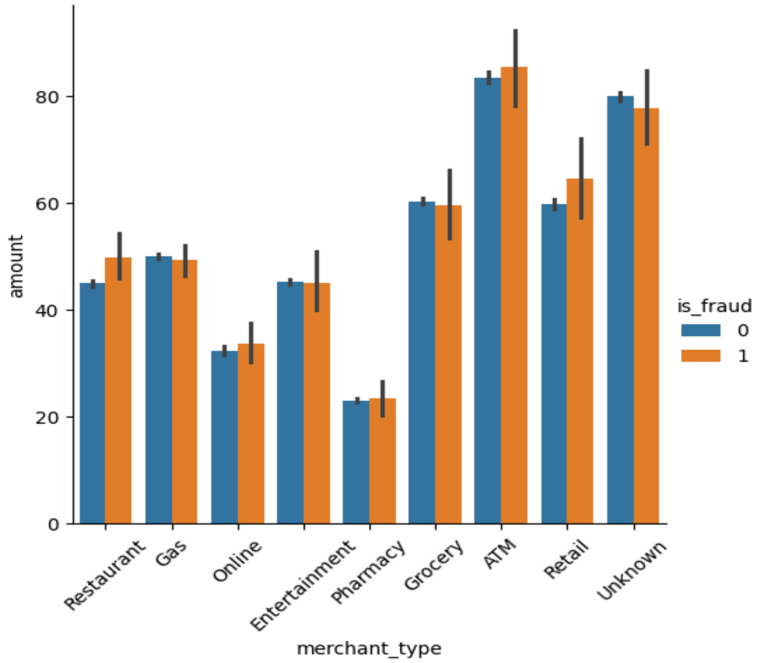


Figure A4. A bar plot displaying the average transaction amount for different merchant types, segmented by fraud status (is_fraud). The plot illustrates that fraudulent transactions (orange bars) generally have a higher average

amount than non-fraudulent transactions (blue bars) across most categories. Error bars represent the standard deviation of transaction amounts.