Toronto Metropolitan University

Master of Science in Data Science and Analytics
MRP - Methodology and Experiments

Optimizing Supervised Machine Learning for Enhanced Transaction Fraud Detection Accuracy

Submitted to:

MRP Supervisor – Dr. Shengkun Xie
MRP Second Reader – Dr. Ceni Babaoglu

Submitted by:

Nguyen Duy Anh Luong (Student ID – 500968520)

July 14, 2025

Aim of Study

In this stage of the project, the goal is to build a supervised machine learning model that can accurately determine whether a credit card transaction is fraudulent (labeled as 1) or legitimate (labeled as 0). This type of binary classification is especially important in the financial industry, where identifying fraud early can help prevent major financial losses and protect both businesses and consumers.

One of the biggest challenges of the project, as discussed in earlier sections, is the imbalance in the dataset. Fraudulent transactions make up only a small portion of all the data. This imbalance can make it seem like a model is performing well based on accuracy alone, when in reality it might be missing a lot of the actual fraud cases. That's why it's crucial to carefully choose the right model, use appropriate evaluation metrics, and apply strategies specifically designed to handle this imbalance effectively.

Data Preprocessing

This part of the analysis relied on two main datasets: train_transaction.csv, which contains detailed transaction data, and train_identity.csv, which holds information related to user identity and devices. These two datasets were combined using the shared TransactionID to create a single dataset for training.

During preprocessing, no features were removed, as this was done intentionally to keep all potentially useful information. For missing values, a simple approach was taken: categorical fields were filled with the placeholder "missing", while missing values in numerical columns were left as "NaN" so that the machine learning models could handle them appropriately later on.

To prepare the data for modeling, categorical features like ProductCD, card4, and DeviceType were converted into numeric format using label encoding. In addition, a new feature called hour_of_day was also created from the TransactionDT column to capture the hour when each transaction occurred, which could be useful in spotting patterns of fraudulent behavior over time. Finally, the processed data was split into training and validation sets using an 80/20 stratified split. This means both sets maintained the same proportion of fraudulent and non-fraudulent transactions, ensuring fair model evaluation across the imbalanced classes.

Handling Class Imbalance

To handle the class imbalance problem, the model was trained using class weighting. Specifically, the scale_pos_weight was set to around 27.58, which reflects the ratio of non-fraudulent to fraudulent transactions in the training set. This helps the model pay more attention to the rare fraud cases without changing the data itself.

Unlike methods such as SMOTE that generate synthetic data, this approach keeps the original dataset intact, maintaining its real-world structure. The validation set remained unchanged throughout the process to ensure that performance metrics reflect how the model would behave on truly unseen data.

Model Selection, Training, and Evaluation

1. XGBoost Classifier

To detect fraudulent transactions, an XGBoost (Extreme Gradient Boosting) model was trained for this binary classification task. XGBoost is a strong choice for this type of structured, tabular data and is especially good at handling imbalanced datasets. The model was set up with a scale_pos_weight of 27.58, which corresponds to the ratio of non-fraudulent to fraudulent transactions in the training data. This helps the model pay more attention to the much smaller number of fraud cases, improving its ability to recognize them accurately.

Model Configuration:

- n_estimators = 100
- max_depth = 6
- learning_rate = 0.1
- scale_pos_weight = 27.58
- subsample = 0.8
- colsample_bytree = 0.8
- random_state: 42

Results on Validation Set:

- Accuracy: 89%
- Recall (Fraud class): 0.81
- Precision (Fraud class): 0.22
- F1-score (Fraud class): 0.35
- ROC-AUC Score: 0.9276
- PR-AUC Score: 0.6187

Confusion Matrix:

	Predicted: Non-Fraud	**Predicted: Fraud**
Actual: Non-Fraud	102,368	11,607
Actual: Fraud	803	3,330

Table 1: XGB - Confusion Matrix

Interpretation:

The model achieved a strong recall of 0.81, meaning it was able to identify most of the fraudulent transactions. Although the precision was lower at 0.22, this is often considered acceptable in this context, since it's typically more important to flag as many fraud cases as possible, even if that leads to some false positives. The ROC-AUC score of 0.9276 indicates strong overall ability to distinguish between fraudulent and non-fraudulent transactions. Meanwhile, the PR-AUC score of 0.6187 reflects consistent precision across different levels of recall.

Curve Analysis:

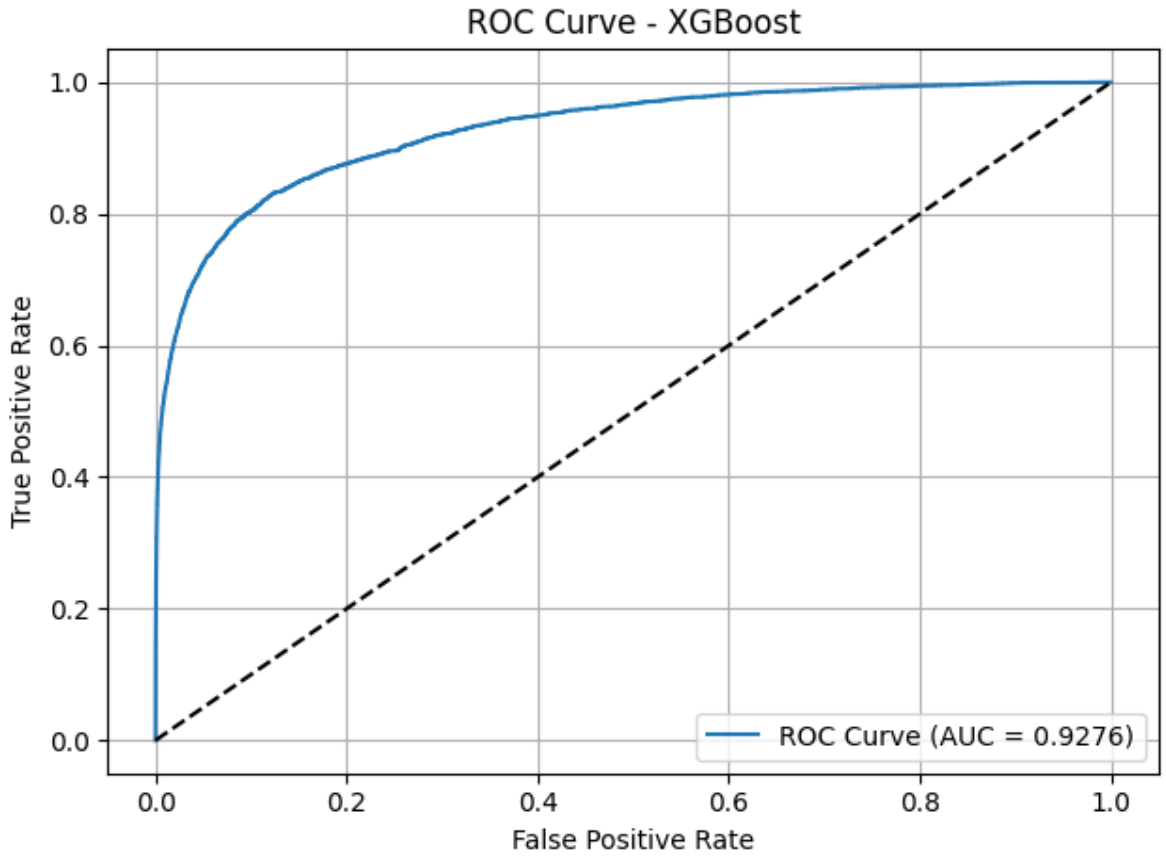

Figure 1: ROC Curve - XGBoost

ROC Curve: The ROC curve rises quickly toward the top-left corner, showing that the model is able to correctly identify a high number of fraud cases while keeping false alarms relatively low. The AUC score of 0.9276 reflects the model's strong overall ability to separate fraudulent from non-fraudulent transactions.

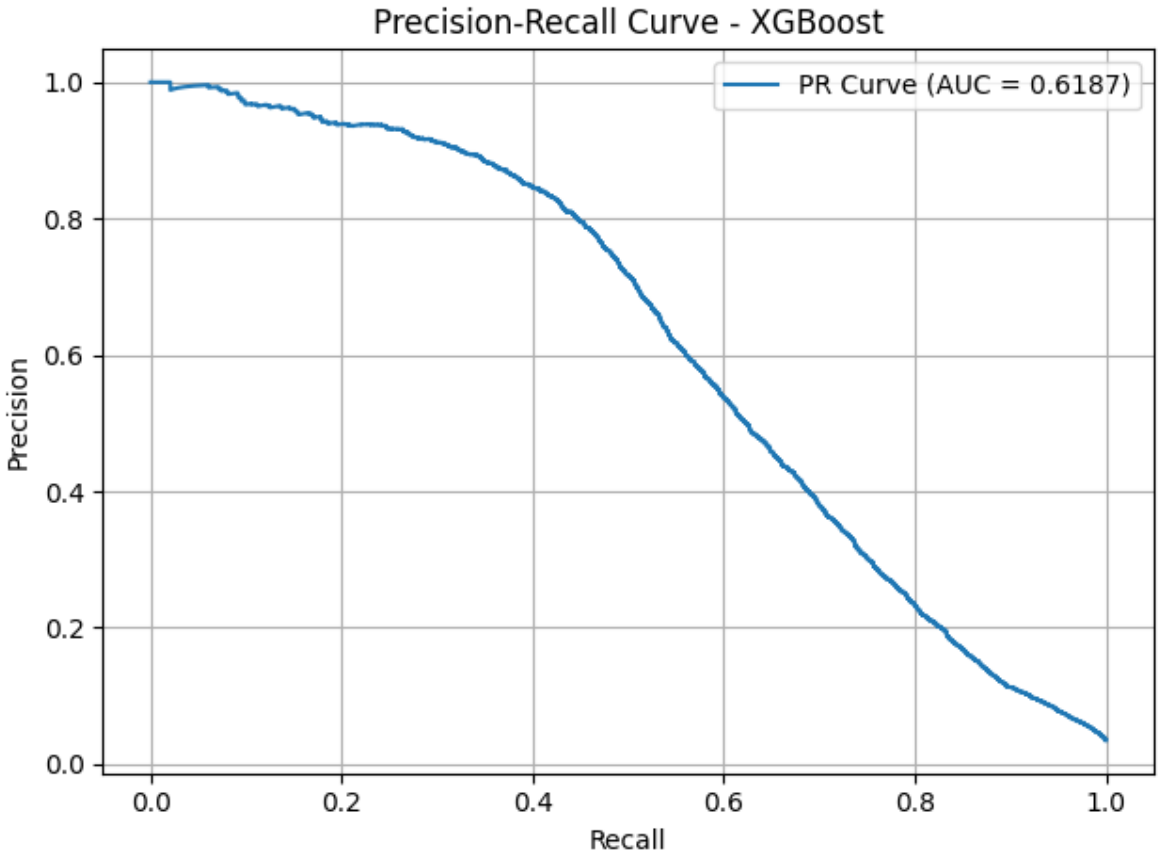

Figure 2: Precision-Recall Curve - XGBoost

Precision-Recall Curve: The precision-recall curve starts off with nearly perfect precision at low recall levels and gradually drops as recall increases. The AUC of 0.6187 is a strong result for imbalanced data, showing the model's effectiveness at maintaining precision while recovering more fraud cases.

Conclusion:

XGBoost turned out to be one of the most effective models in this project for detecting fraud. It was able to identify a large share of the fraudulent transactions, with a recall of 0.81, and showed strong overall performance in distinguishing between fraud and non-fraud cases, as reflected by its ROC-AUC score of 0.9276. While the precision was lower at 0.22, that's often an acceptable trade-off in fraud detection. With further threshold tuning or integration into an ensemble system, XGBoost could serve as a foundation for real-time fraud detection pipelines.

2. Random Forest Classifier

Random Forest is a reliable ensemble method that works by building many decision trees and combining their results to make more accurate predictions while avoiding overfitting. It's especially useful for fraud detection because it can pick up on complex patterns in the data and isn't easily thrown off by noise. In this experiment, the model was set up with class_weight='balanced' to account for the large difference in the number of fraud versus non-fraud transactions, helping it focus more on the minority class.

Model Configuration:

- n_estimators = 100
- max_depth = 15
- class_weight = 'balanced'
- n_jobs = -1 (parallelized training)
- random_state: 42

Results on Validation Set:

- Accuracy: 94%
- Recall (Fraud class): 0.68
- Precision (Fraud class): 0.33
- F1-score (Fraud class): 0.45
- ROC-AUC Score: 0.9122
- PR-AUC Score: 0.5856

Confusion Matrix:

	Predicted: Non-Fraud	**Predicted: Fraud**
Actual: Non-Fraud	108,314	5,661
Actual: Fraud	1,323	2,810

Table 2: Random Forest - Confusion Matrix

Interpretation:

The Random Forest model showed solid performance, detecting 68% of fraud cases with a precision of 33%. While it does produce some false positives, this is generally acceptable in fraud detection, where it's better to be cautious than to miss fraud. Compared to simpler models like Logistic Regression, it improves recall and overall discrimination. Although its recall is slightly lower than XGBoost's, it generates fewer false alarms, providing a more balanced trade-off between sensitivity and specificity.

Curve Analysis:

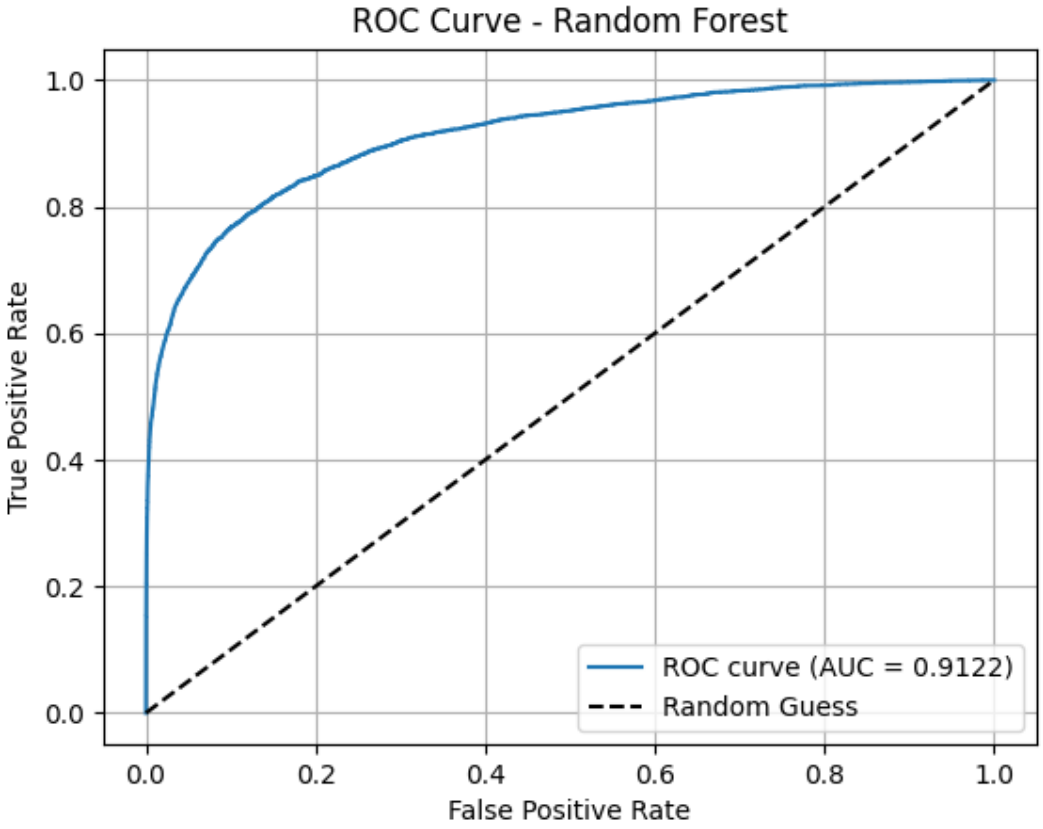

Figure 3: ROC Curve - Random Forest

ROC Curve: The ROC curve for Random Forest shows a strong separation between the classes, with an area under the curve (AUC) of 0.9122. This indicates that the model is highly effective at ranking fraudulent vs. non-fraudulent transactions across different thresholds.

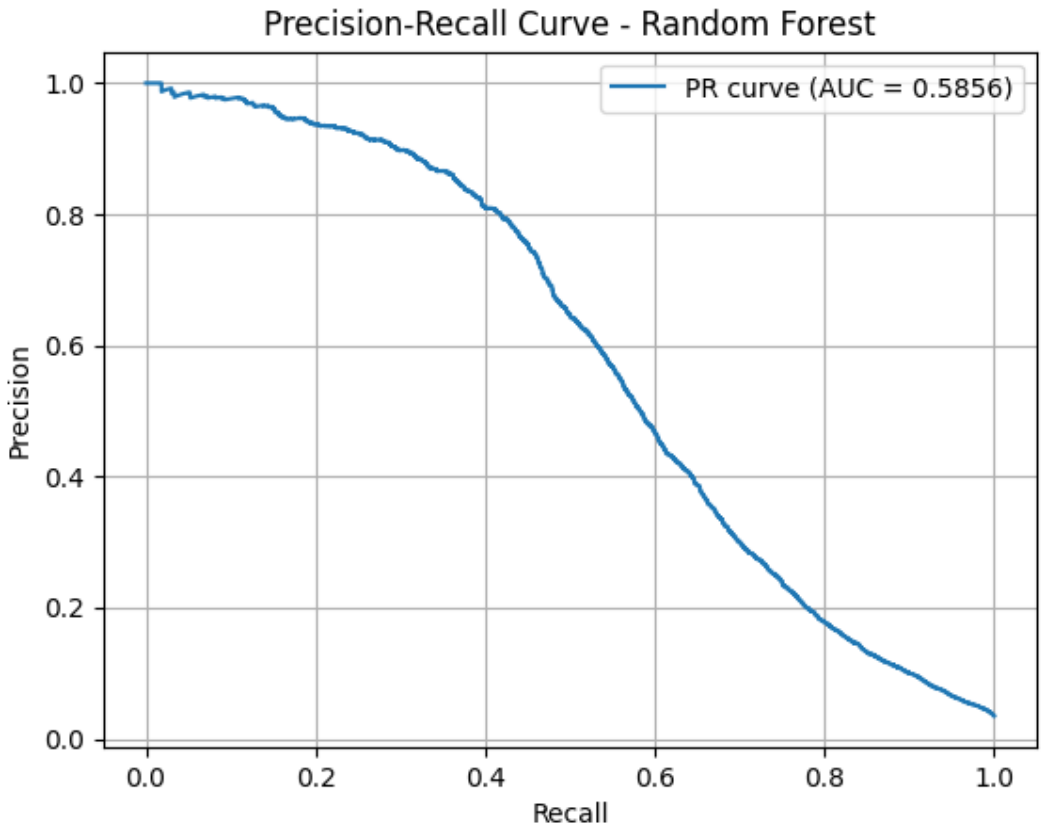

Figure 4: Precision-Recall Curve - Random Forest

Precision-Recall Curve: The PR-AUC of 0.5856 further highlights the model's capability to maintain a balance between detecting fraud (recall) and reducing false positives (precision), which is especially critical in imbalanced classification problems like fraud detection.

Conclusion:

Random Forest is a reliable and effective model for fraud detection. Its ability to detect a majority of fraudulent transactions while maintaining an acceptable level of false positives makes it suitable for real-world applications, particularly in environments that require both accuracy and interpretability. Its strong results across key metrics, such as high ROC-AUC and solid PR-AUC, show that Random Forest can be used as part of fraud prevention systems, particularly when combined with human review or other models in an ensemble.

3. Logistic Regression

Logistic Regression was used as a baseline model to classify credit card transactions as either fraudulent (1) or non-fraudulent (0) using the preprocessed feature set. Due to its simplicity and interpretability, it serves as a benchmark for evaluating more complex models. Class imbalance was addressed by setting class_weight="balanced", and missing values were filled in using the median strategy to keep the data compatible with the model.

Model Configuration:

- Solver: liblinear
- class_weight = 'balanced'
- Max Iterations: 200
- random_state: 42

Results on Validation Set:

- Accuracy: 77%
- Recall (Fraud class): 0.69
- Precision (Fraud class): 0.10
- F1-score (Fraud class): 0.18
- ROC-AUC Score: 0.7959
- PR-AUC Score: 0.1772

Confusion Matrix:

	Predicted: Non-Fraud	**Predicted: Fraud**
Actual: Non-Fraud	88,579	25,396
Actual: Fraud	1,296	2,837

Table 3: Logistic Regression - Confusion Matrix

Interpretation:

The model achieves a high recall (0.69) on the fraud class, meaning it captures a majority of fraudulent transactions. However, the precision is extremely low (0.10), indicating that only 1 in 10 fraud predictions is correct. This high false positive rate can cause alert fatigue and unnecessary operational costs.

While the ROC-AUC score of 0.7959 suggests the model is moderately effective at distinguishing between fraud and non-fraud classes, the PR-AUC score of 0.1772 reveals a bigger problem: the model struggles to keep precision high as it tries to identify more fraud cases, which is a major challenge when working with such imbalanced data.

Curve Analysis:

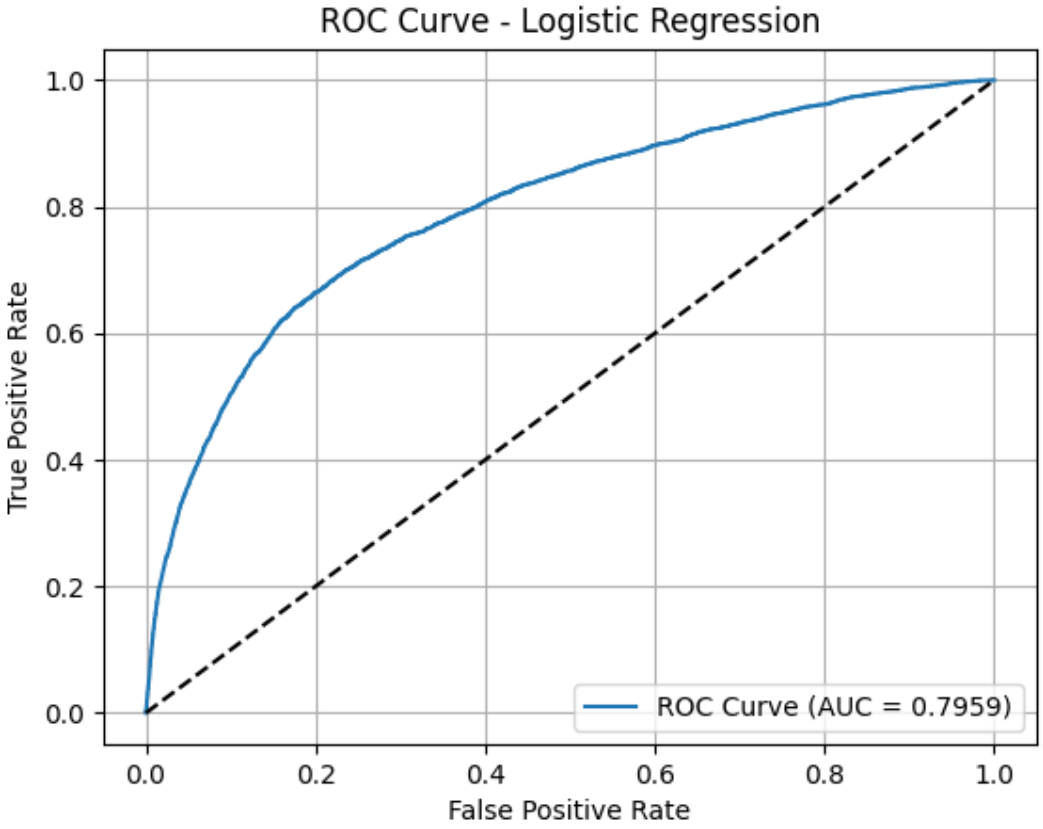

Figure 5: ROC Curve - Logistic Regression

ROC Curve: The ROC curve shows the trade-off between the true positive rate (sensitivity) and false positive rate. For Logistic Regression, the curve lies above the diagonal line, showing it has some ability to tell fraud from non-fraud, but it's not strong enough to be considered highly reliable. The AUC score of 0.7959 reflects this moderate level of separation.

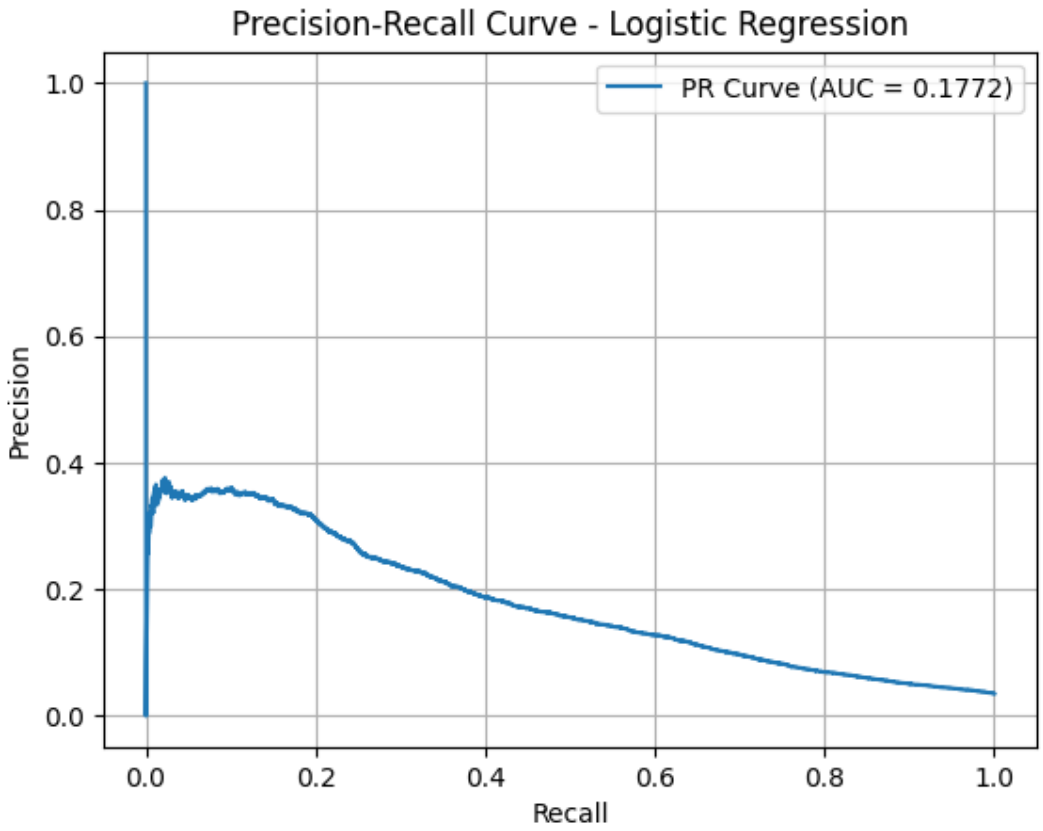

Figure 6: Precision-Recall Curve - Logistic Regression

Precision-Recall Curve: This curve is especially useful when dealing with imbalanced data. It shows that as the model tries to catch more fraud cases (higher recall), its precision drops quickly, meaning many of the flagged cases aren't actually fraud. The low PR-AUC of 0.1772 highlights that, although the model detects some fraud, it also produces a lot of false alerts.

Conclusion:

Logistic Regression provides a quick, interpretable starting point for fraud detection. However, its performance, particularly the low precision and PR-AUC, makes it unsuitable for utilized in environments where false positives are costly. More advanced methods such as Random Forest and XGBoost demonstrate stronger fraud detection capabilities and better balance between precision and recall.

4. Artificial Neural Network (ANN) - Shallow 2-layer

To classify credit card transactions as fraudulent (1) or non-fraudulent (0), a feedforward artificial neural network (ANN) was trained on preprocessed transactional and identity data. This ANN serves as a relatively simple deep learning model for benchmarking more complex architectures.

Model Architecture and Configuration:

- Model Type: Feedforward Neural Network (ANN)
- Input Features: Scaled numerical and label-encoded categorical features from merged datasets
- Hidden Layers:
  - Dense(128), ReLU activation, Dropout(0.3)
  - Dense(64), ReLU activation, Dropout(0.3)
- Output Layer: Sigmoid activation (for binary classification)
- Loss Function: Binary Cross-Entropy
- Optimizer: Adam (learning rate = 0.001)
- Training Configuration: 20 epochs, batch size = 512, with early stopping based on validation loss

Results on Validation Set:

- Accuracy: 98%
- Recall (Fraud class): 0.43
- Precision (Fraud class): 0.91
- F1-score (Fraud class): 0.59
- ROC-AUC Score: 0.9160
- PR-AUC Score: 0.6452

Confusion Matrix:

	Predicted: Non-Fraud	**Predicted: Fraud**
Actual: Non-Fraud	113,801	174
Actual: Fraud	2,337	1,796

Table 4: ANN - Confusion Matrix

Interpretation:

The ANN model achieved high precision on fraud cases (0.91), meaning most flagged transactions were truly fraudulent. Although recall is moderate at 0.43, the model still identifies nearly half of all fraud cases. With an F1-score of 0.59, it offers one of the best trade-offs among the tested models, making it a strong option when minimizing false positives is a priority without fully compromising fraud detection.

Curve Analysis:

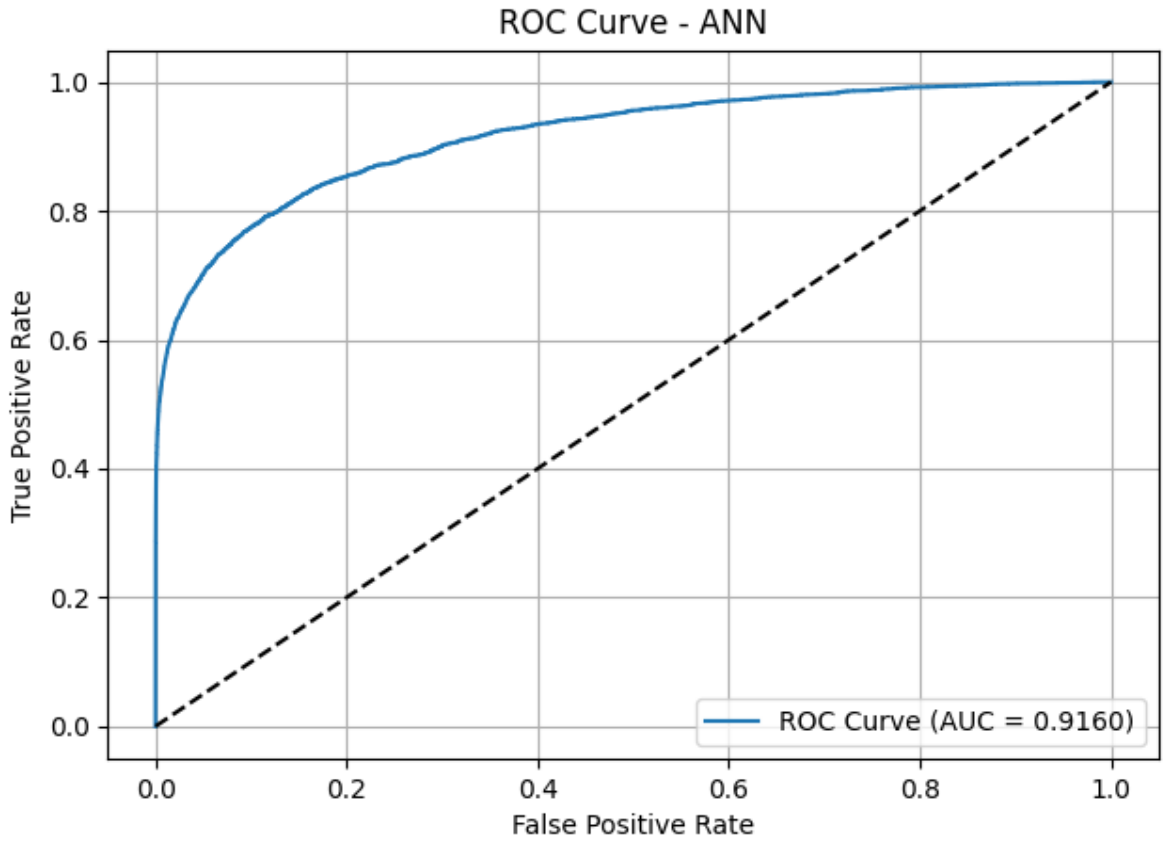

Figure 7: ROC Curve - ANN

ROC Curve: The ROC curve shows strong class separation, with a ROC-AUC of 0.9160, indicating the model reliably distinguishes between fraud and non-fraud across various thresholds.

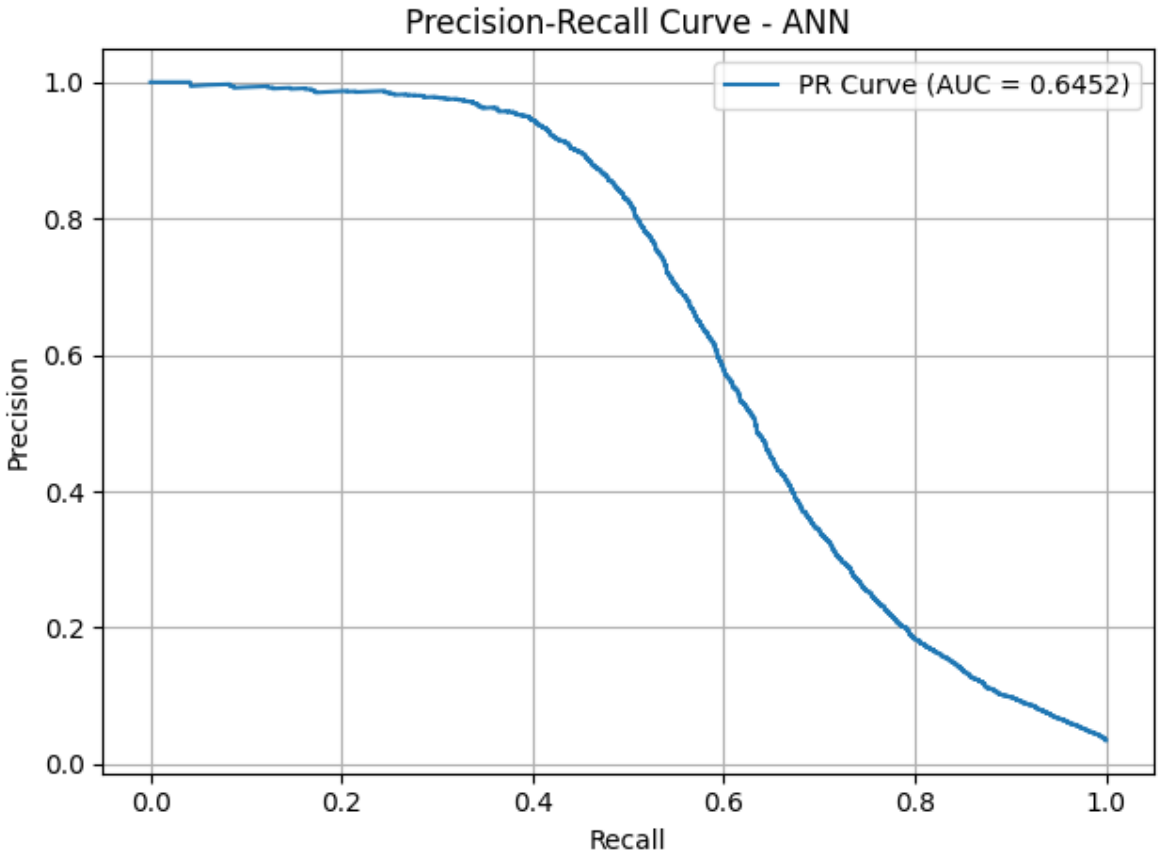

Figure 8: Precision-Recall Curve - ANN

Precision-Recall Curve: In cases of class imbalance, the precision-recall curve gives a clearer picture of model performance. With a PR-AUC of 0.6452, the ANN does a great job of maintaining high precision even as it picks up more fraud cases. This is the highest PR-AUC of any model tested so far, highlighting its strength in real-world fraud detection.

Conclusion:

The ANN demonstrates excellent potential as a high-precision model for fraud detection. It is especially valuable in applications where false positives must be minimized. With further tuning, such as threshold adjustment, more training epochs, or deeper architecture, its recall could potentially be improved without sacrificing its strong precision.

5. Deep Neural Network (DNN) - Deeper 3+ layers

To classify credit card transactions as fraudulent (1) or non-fraudulent (0), a deep feed-forward neural network (DNN) was trained on preprocessed transactional and identity features. The architecture includes multiple hidden layers with batch normalization and dropout for regularization and employs focal loss to specifically address the class imbalance challenge.

Model Architecture and Configuration:

- Model Type: Deep Neural Network (DNN)
- Input Features: Preprocessed numerical features from the merged dataset
- Hidden Layers:
  - Layer 1: Dense(512) → LeakyReLU → BatchNormalization → Dropout(0.4)
  - Layer 2: Dense(256) → LeakyReLU → BatchNormalization → Dropout(0.4)
  - Layer 3: Dense(128) → LeakyReLU → BatchNormalization → Dropout(0.3)
- Output Layer: Dense(1) with Sigmoid activation
- Loss Function: Focal Loss ($\gamma = 2.0$, $\alpha = 0.25$)
- Optimizer: Adam (learning rate = 0.001)
- Class Imbalance Handling: Focal Loss + Probability Thresholding (0.3)
- Training Configuration: 20 epochs, batch size = 1024

Results on Validation Set:

- Accuracy: 98%
- Recall (Fraud class): 0.51
- Precision (Fraud class): 0.82
- F1-score (Fraud class): 0.63
- ROC-AUC Score: 0.9225
- PR-AUC Score: 0.6582

Confusion Matrix:

	Predicted: Non-Fraud	**Predicted: Fraud**
Actual: Non-Fraud	113,498	477
Actual: Fraud	2,016	2,117

Table 5: DNN - Confusion Matrix

Interpretation:

Deep neural networks (DNN) provide a strong balance between detecting fraud and minimizing false alarms. With a precision of 0.82, most of the transactions it flags as fraud are correct. It also achieves a recall of 0.51, meaning it successfully identifies over half of the actual fraud cases. The F1-score of 0.63 further reflects this balanced performance, making the model well-suited for real-world risk management scenarios where both accuracy and coverage matter.

Curve Analysis:

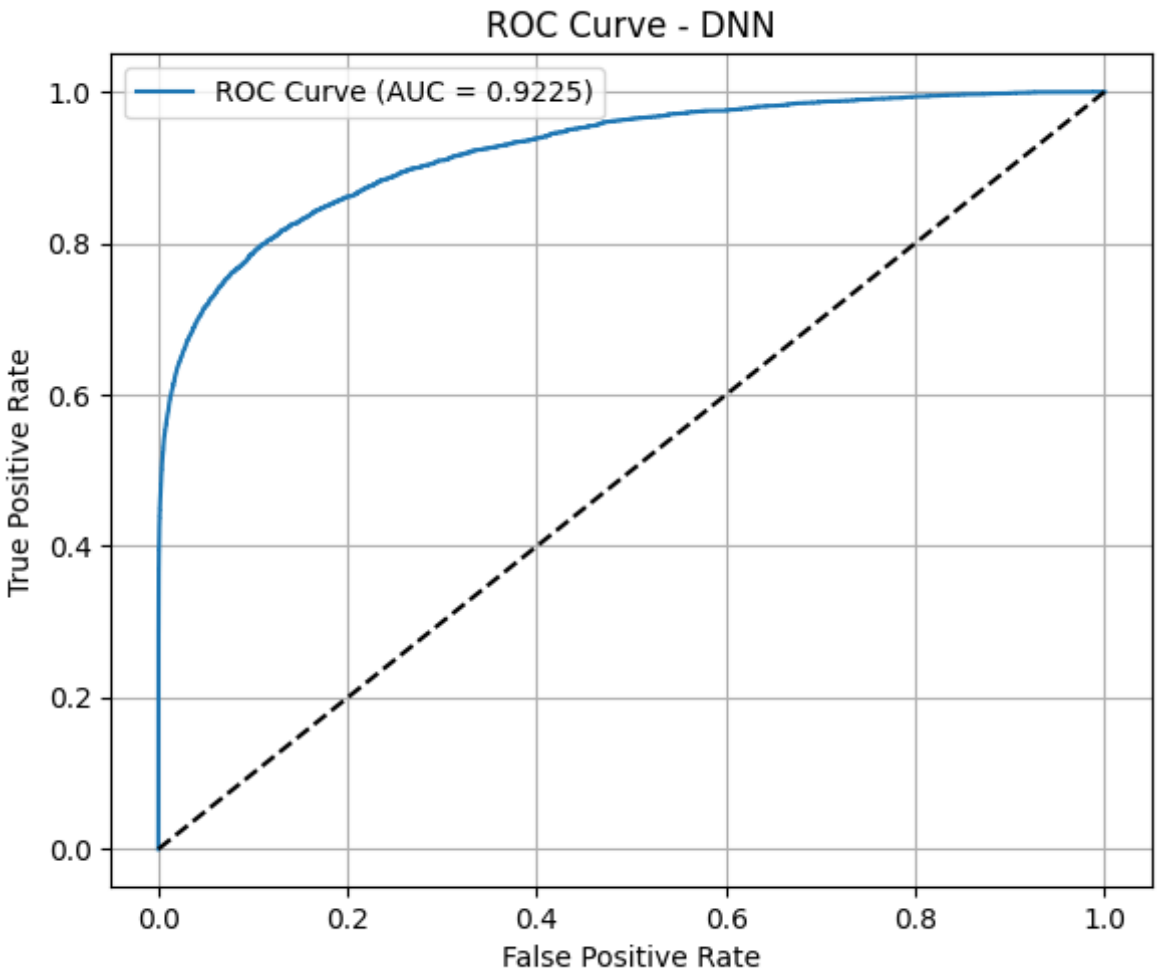

Figure 9: ROC Curve - DNN

ROC Curve (AUC = 0.9225): The ROC curve shows strong class separation, with a steep rise toward the top-left corner. This indicates the model is effective at distinguishing between fraudulent and non-fraudulent transactions across various thresholds.

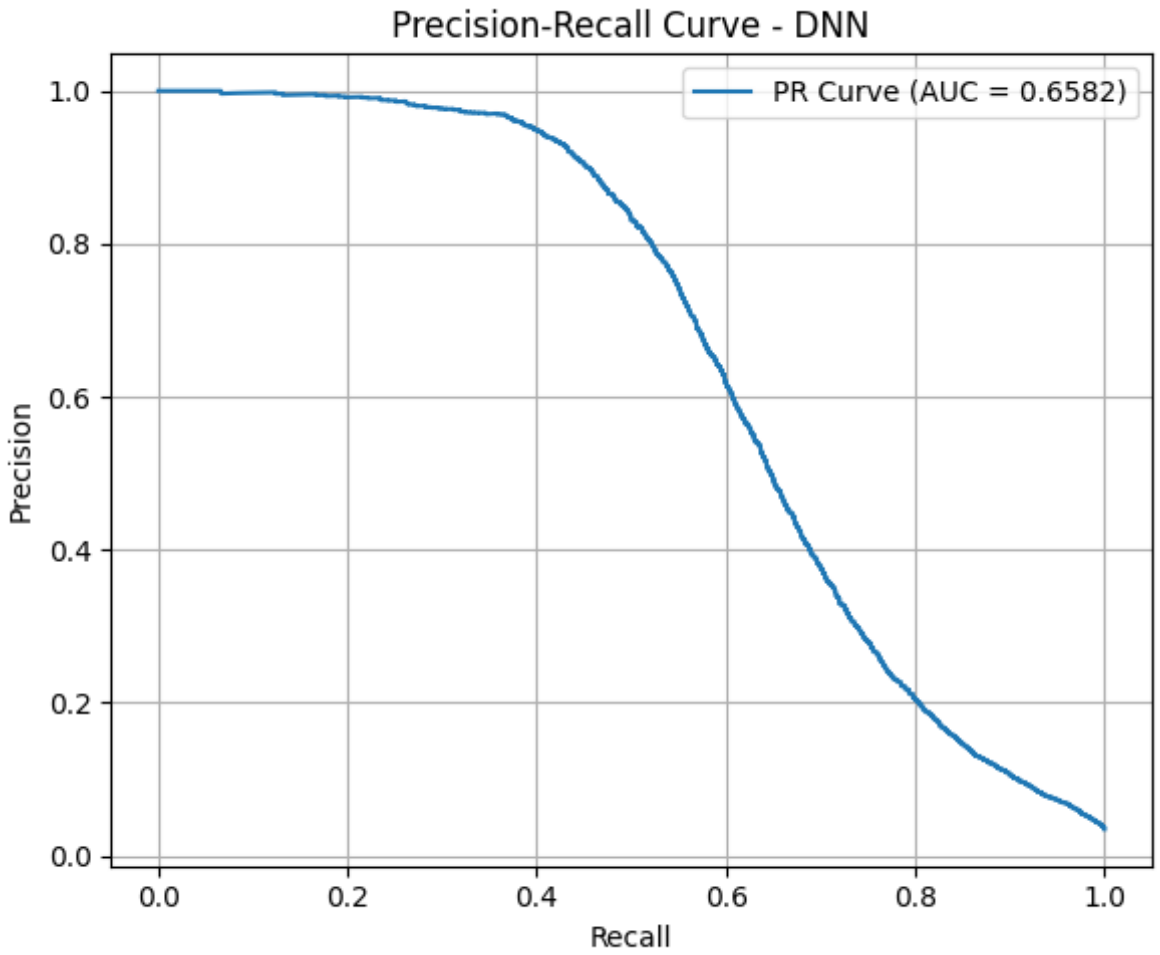

Figure 10: Precision-Recall Curve - DNN

Precision-Recall Curve (PR-AUC = 0.6582): The precision-recall curve highlights the model's ability to maintain high precision as recall increases, which is a key strength in fraud detection, where false positives can be disruptive and actual fraud cases are rare.

Conclusion:

The DNN model performs well across the board, with high accuracy, a strong ROC-AUC score, and excellent precision on fraud cases, making it a solid candidate for real-world fraud detection systems. Its use of focal loss helps it handle imbalanced data effectively, and it shows good resistance to overfitting. After 20 training epochs, the model delivers consistently strong results, and there's still room for improvement through further tuning or feature engineering. Overall, DNN stands out as one of the most balanced models in this project.

Experimental Results - Summary

Model	Accuracy	Recall	Precision	F1-Score	ROC-AUC	PR-AUC
Logistic Regression (LR)	77%	0.69	0.10	0.18	0.7959	0.1772
Random Forest (RF)	94%	0.68	0.33	0.45	0.9122	0.5856
XGBoost (XGB)	89%	0.81	0.22	0.35	0.9276	0.6187
Artificial Neural Network (ANN)	98%	0.43	0.91	0.59	0.9160	0.6452
Deep Neural Network (DNN)	98%	0.51	0.82	0.63	0.9225	0.6582

Table 6: Performance Comparison Across Models

Best Overall Model - Deep Neural Network (DNN): With excellent accuracy (98%), strong precision (0.82), balanced F1-score (0.63), and high ROC-AUC and PR-AUC, the DNN offers the most reliable solution. It is especially suitable for production systems that require both accurate fraud detection and effective control over false alarms.

Alternative Option - XGBoost: If prioritizing maximum fraud recall, even at the cost of precision, XGBoost remains a leading choice with the highest recall (0.81) and best ROC-AUC (0.9276).

Feature Importance and Model Interpretation

A. Interpretation of SHAP Analysis - XGBoost Classifier

To better understand which features influence the XGBoost model's predictions for fraud detection, SHAP (SHapley Additive exPlanations) values were computed. SHAP helps explain how each feature contributes to the model's decisions, both overall and for individual predictions, making it a valuable tool for understanding complex models.

Global Feature Importance:

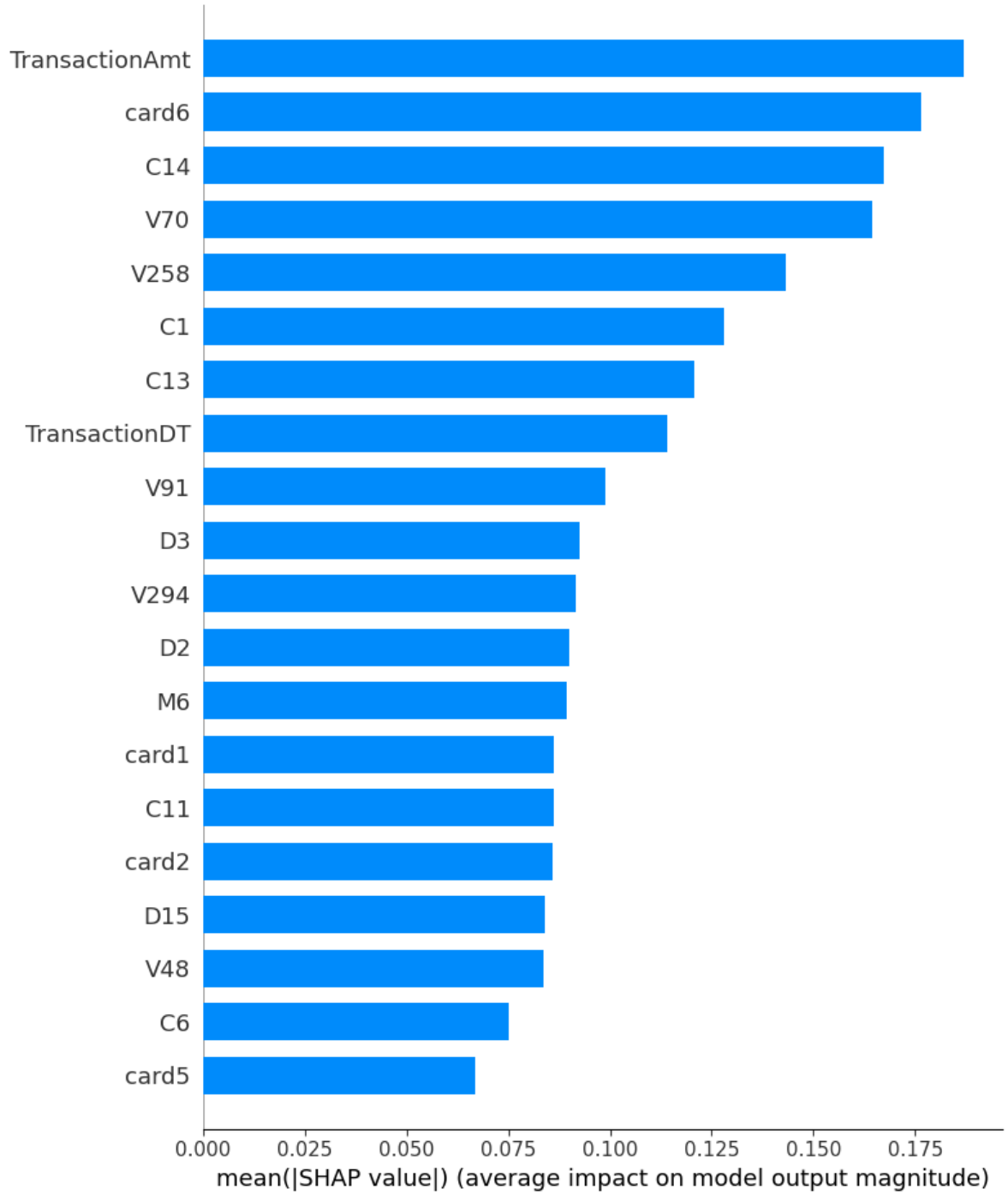

Figure 11: Global Feature Importance - Bar Plot

The SHAP bar plot ranks features based on their average contribution to the model's output across all validation samples. From the plot, the most influential features in predicting fraud include:

- **TransactionAmt** - The transaction amount is the strongest predictor. Larger or unusual amounts tend to raise the model's fraud probability.
- **card6** - Likely reflects card type (e.g., debit, credit, etc.). Certain types may be more associated with fraudulent behavior.
- **C14, C1, C13** - These anonymized features (often linked to customer ID, address match, or transaction count) show strong separation power.
- **V70, V258, V91, V294** - These V* engineered features carry behavioral or device-related information.
- **TransactionDT** – Represents the time of transaction. Time-based patterns can signal fraud trends (e.g., night-time activity).
- **D features** – Often indicate time deltas between various customer activities and help catch suspicious timing anomalies.

Together, these features suggest that the model has learned a combination of transaction size, card/device identity, and temporal behavior to distinguish between fraud and non-fraud transactions.

Local Explanation:

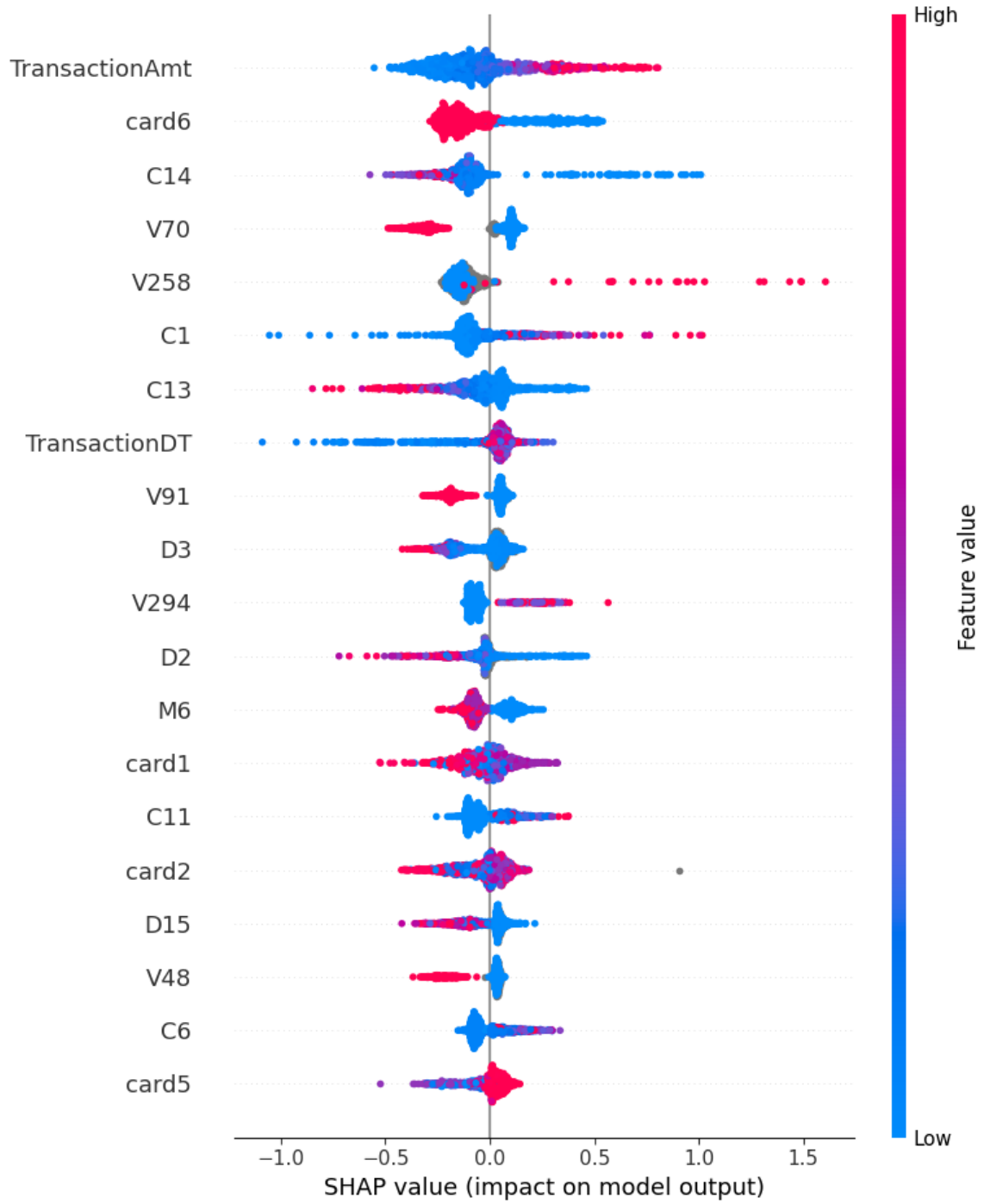

Figure 12: Local Explanation - Beeswarm Plot

The SHAP beeswarm plot gives a closer look at how individual feature values influence the model's decisions for each transaction.

- When a point appears on the right side of the plot (positive SHAP values), it means that particular feature value pushes the model toward predicting fraud. Points on the left (negative values) push the model toward predicting non-fraud.
- For example, higher values of **TransactionAmt** are usually linked to a greater chance of fraud. This is shown by red points clustering on the right side of the plot. In contrast, smaller transaction amounts tend to reduce the model's fraud score.
- Features like `card6` and `C14` have different impacts depending on their specific values, with some categories or ranges increasing the risk of fraud more than others.

This kind of insight helps not only with debugging the model but also with building trust, especially for stakeholders who want to understand the reasoning behind flagged transactions.

Overall, the SHAP analysis reveals that the XGBoost model relies on a meaningful combination of transaction metadata, cardholder identifiers, and temporal features to make its predictions. This transparency ensures that fraud analysts can trust and act upon the model's predictions with confidence.

B. Feature Importance Analysis - Random Forest

To better understand how the Random Forest model makes its decisions, feature importance scores from the built-in `"feature_importances_attribute"` were examined. These scores indicate how much each feature contributes to the model's predictions by measuring its impact on reducing uncertainty across all the decision trees.

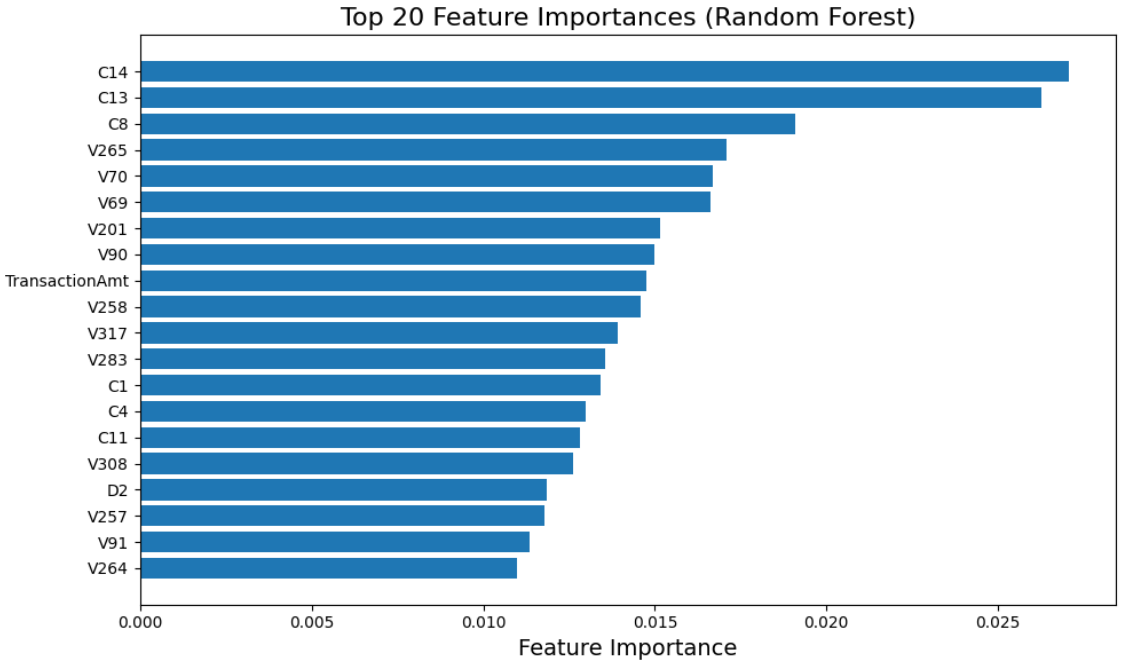

Figure 13: Random Forest - Top 20 Feature Importances

Key Findings:

- Features like **C14**, **C13**, and **C8** had the highest influence on the model's predictions. While these are anonymized, they likely represent patterns tied to customer behavior or transaction structure, making them strong indicators of fraud.
- Variables such as **TransactionAmt** and engineered features like **V258**, **V201**, and **D2** also ranked highly, which aligns with common fraud patterns involving unusual amounts and suspicious timing.
- What stands out is the broad distribution of importance across more than 20 features. This suggests that the Random Forest model draws insights from many different signals, rather than depending on just a few key predictors.

Overall, the feature importance plot provides a quick view of which inputs the Random Forest model relies on most when detecting fraud. While it doesn't offer detailed explanations like SHAP, it's still useful for identifying key features, understanding the model, and communicating results clearly. Features such as **C14**, **V258**, and **TransactionAmt** contribute the most, offering insights that can guide further feature engineering and support ongoing improvements to the fraud detection systems.

GitHub Repository

All codes and visualizations for this Methodology and Experiments stage are available at the following GitHub repository:
Link: `https://github.com/nguyenduyanhluong/TMU-MRP-2025/tree/main/methodology_and_experiments`