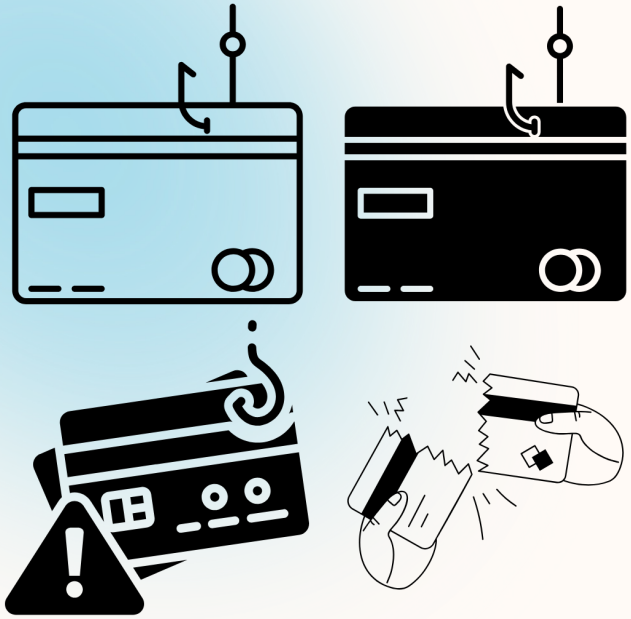
Harnessing Transformers for Fraud Detection

- Tackling Imbalanced Data and Outperforming Baselines



Fengyuan Shen
Jinhu Sun

DS542 Deep Learning
December 6, 2024

Table of contents

What Is Credit Card Fraud?

A type of identity theft that **happens when someone steals your credit card information** to make fraudulent purchases or transfer funds.

01

Introduction

Introduction

01. Overview

- **Financial fraud**: A growing threat to global economic stability.
- **Challenges**:
  - Highly **imbalanced data**: Fraudulent transactions make up only 0.172%.
  - Complex, non-linear patterns in transaction data.

02. Dataset Summary

- **Dataset**: Credit Card Fraud Detection dataset (Kaggle).
- **Size**: 284,807 transactions, 492 fraudulent.
- **Features**:
  - PCA-transformed variables (V1–V28).
  - Time and Amount are raw features.

03. Objectives

- Build an **effective fraud detection model**.
- Improve **recall for minority (fraudulent) class**.
- Balance **minority class detection** without sacrificing majority class performance.

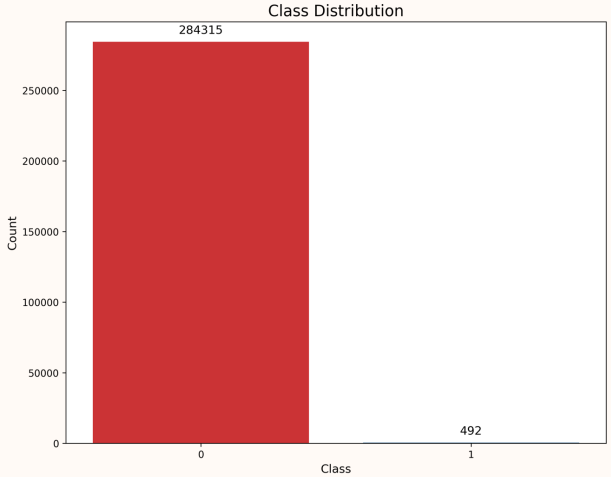
02

Methodology

Methodology

01. Data Preprocessing

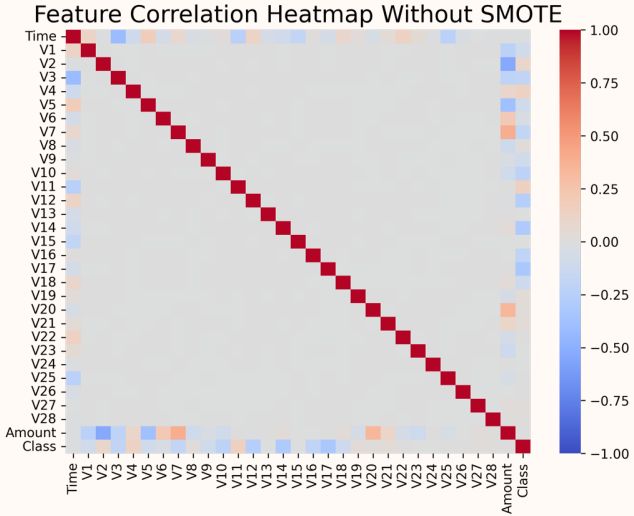
- **Training Set Splitting**: 70:30, sorted by **Time** to mimic real-world fraud detection.
- **Standardization**:
  - Both Time and Amount were standardized separately for **training** and **testing** datasets to prevent data leakage.
- **SMOTE for Imbalance Handling**:
  - Balanced classes in training set (1:1 ratio).
  - Generate synthetic samples for the minority class by interpolating between existing samples.
  - Result: Improved minority class representation.

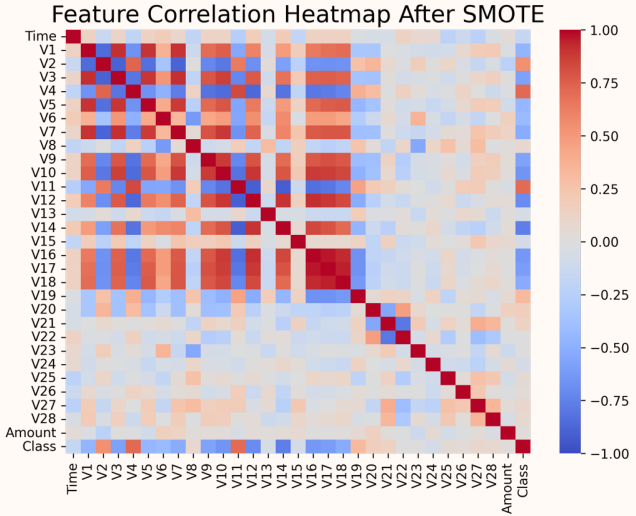


Methodology

02. Correlation Analysis

- **Purpose**: To investigate the relationships between features and the target variable.
- **Challenge**: In imbalanced datasets, correlation coefficients can be misleading due to the scarcity of minority class samples.
- **Impact of SMOTE**:
  - SMOTE increases the proportion and diversity of minority class samples.
  - Enhanced correlations reveal more distinctive patterns between features and the target variable.
  - Results help models better discern key relationships for fraud detection.





Methodology

03. Model Comparison

- **Baseline Models**:
  - Logistic Regression
  - Random Forest
- **Deep Learning Models**:
  - Multi-Layer Perceptron (MLP).
  - LSTM for sequence modeling.
  - Transformer leveraging self-attention.

Model	Architecture	Key Layers	Loss	Optimizer	Learning Rate	Epochs
Logistic Regression	Single linear layer + sigmoid	Input dim -> 1	BCE	Adam	0.001	30
Random Forest	Ensemble of 100 trees	n_estimators = 100	N/A	N/A	N/A	N/A
MLP	Fully connected network	fc1: 128; fc2: 64	BCE	Adam	0.001	30
LSTM	2-layer LSTM + linear output	hidden_dim: 64; num_layers: 2	BCE or Focal	Adam	0.001	30
Transformer	2-layer encoder; multi-head attention	d_model: 256; heads: 4	BCE or Focal	AdamW	0.001	30

Methodology

04. Loss Functions

- **Binary Cross-Entropy (BCE) Loss**:
  - Standard loss for binary classification:

$$L_{BCE}(p, y) = -[y\ log(p)\ +\ (1\ -\ y)\ log(1\ -\ p)]$$

  - **Limitations**:
    - Treats all classes equally, ignoring class imbalance.
    - Leads to bias toward the majority class, as misclassifying rare cases has minimal impact on overall loss.
- **Focal Loss**:
  - Designed to address class imbalance by extending BCE loss:

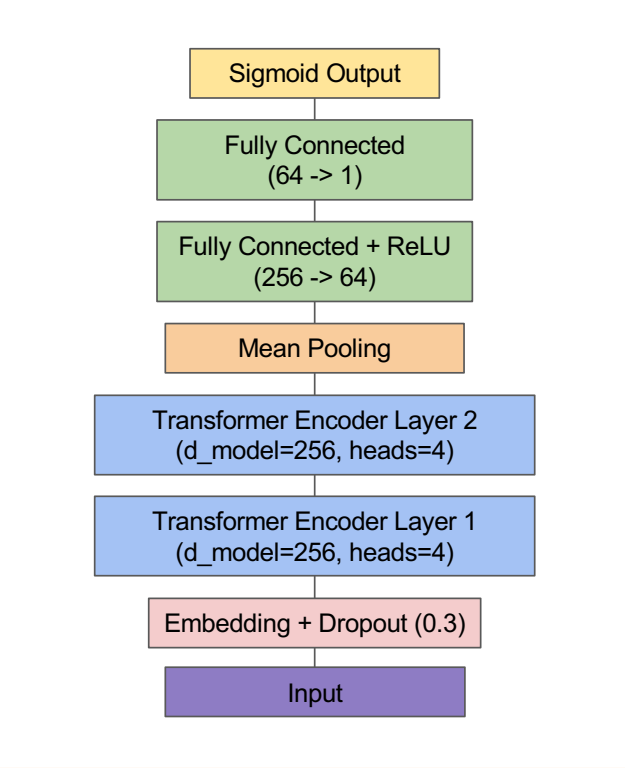
$$L_{focal}(p, y) = -[\alpha\ y\ (1-p)^{\gamma}\ log(p)\ +(1-\alpha)\ (1\ -\ y)\ p^{\gamma}\ log(1\ -\ p)]$$

  - **Key Parameters**:
    - **α**: Adjusts the weight between positive and negative classes (e.g., α=0.5 for balance).
    - **γ**: Focuses on hard-to-classify examples, with larger values emphasizing more difficult samples.
  - **Special Case**:
    - When α=0.5 and γ=0, Focal Loss simplifies to BCE Loss.

Methodology

05. Innovation: Transformer with Focal Loss

- **Why Transformer**:
  - Captures both local and global patterns efficiently.
- **Why Focal Loss**:
  - Focuses on hard-to-classify fraud cases.
  - Reduces the dominance of majority class.
  - **Result**: Enhanced recall for fraudulent transactions.



06. Evaluation Metrics

- **Recall**:
  - **Definition**: Measures the proportion of actual fraud cases correctly identified.
  - **Importance**: Prioritizes identifying rare fraudulent transactions over overall accuracy.
- **ROC AUC Score**:
  - **Definition**: Evaluates the model's ability to rank positive instances (fraud) higher than negative ones (non-fraud) across varying classification thresholds.
  - **Benefits**: Provides a holistic view of the model's discriminative power.

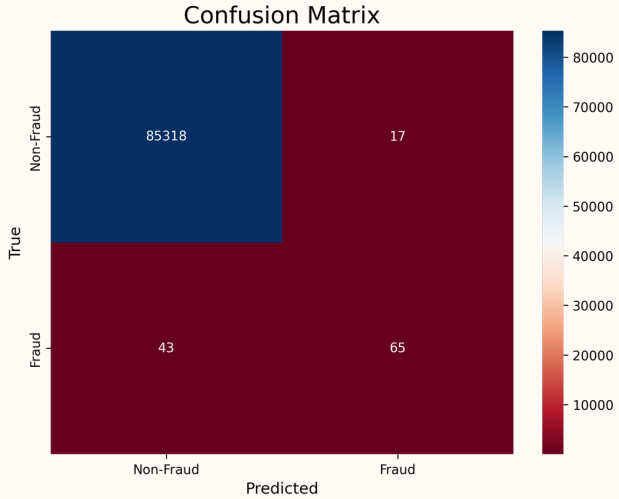
03

Experiments and Results

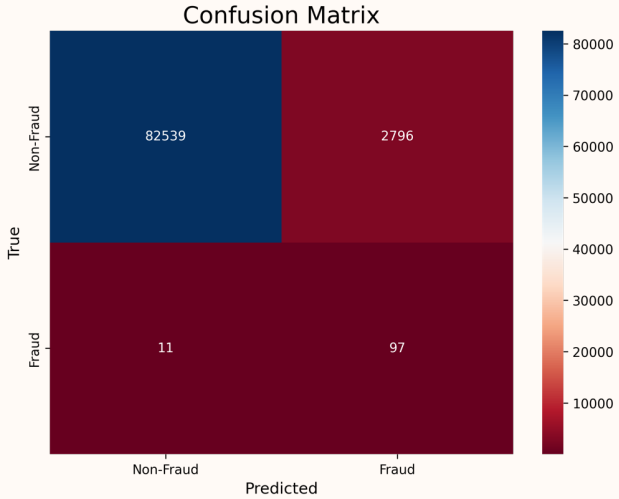
Experiments and Results

01. Effect of SMOTE on Logistic Regression

- **Goal**: Highlight the impact of addressing class imbalance.
- **Findings**:
  - Without SMOTE: High accuracy but poor fraud detection (Recall = 0.60, ROC AUC = 0.80).
  - With SMOTE: Significant improvement in fraud recall (Recall = 0.90, ROC AUC = 0.93).
- **Trade-off**: Increased false positives for non-fraud cases.



Logistic Regression Without SMOTE



Logistic Regression With SMOTE

Experiments and Results

02. Model Performance Comparison

- **Models Evaluated**: Logistic Regression, Random Forest, MLP, LSTM, Transformer.
- **Key Results**:
  - Logistic Regression (after SMOTE): High fraud recall (0.90).
  - Transformer: Achieved balance with fraud recall (0.89) and overall performance (ROC AUC = 0.91).
- **Insight**: Transformer effectively captures nuanced patterns but trades off some non-fraud accuracy.

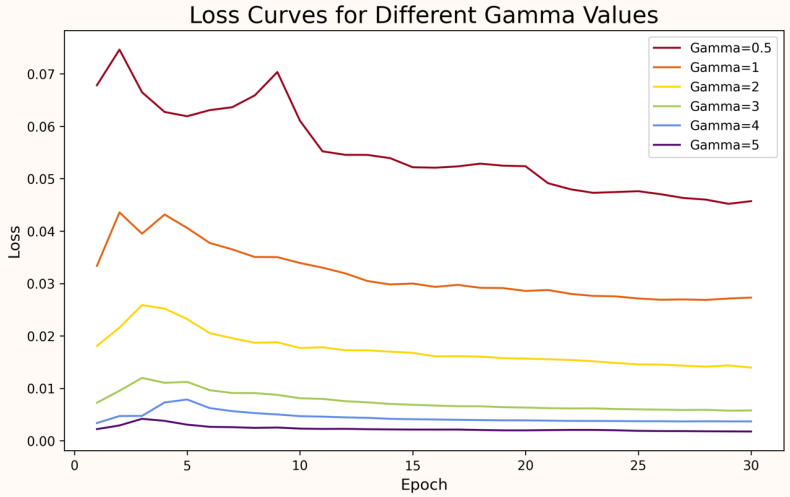
Model	Recall (Fraud)	Recall (Non-Fraud)	ROC AUC Score
Logistic Regression	**0.90**	**0.97**	**0.93**
Random Forest	0.76	1.00	0.88
MLP	0.79	1.00	0.89
LSTM	0.77	1.00	0.88
Transformer	**0.89**	**0.94**	**0.91**

Performance of Various Models (Trained with BCELoss and SMOTE)

Experiments and Results

03. Impact of Focal Loss on Transformer

- **Experiment**: Tuning γ (Focal Loss) to improve fraud detection.
- **Key Findings**:
  - γ=2: Optimal trade-off between fraud recall (0.917) and non-fraud performance (0.931).
  - Lower γ (e.g., 0.5): Focuses aggressively on hard samples but sacrifices non-fraud accuracy.
- **Comparison**: Achieved similar ROC AUC to Logistic Regression while exceeding its fraud recall.



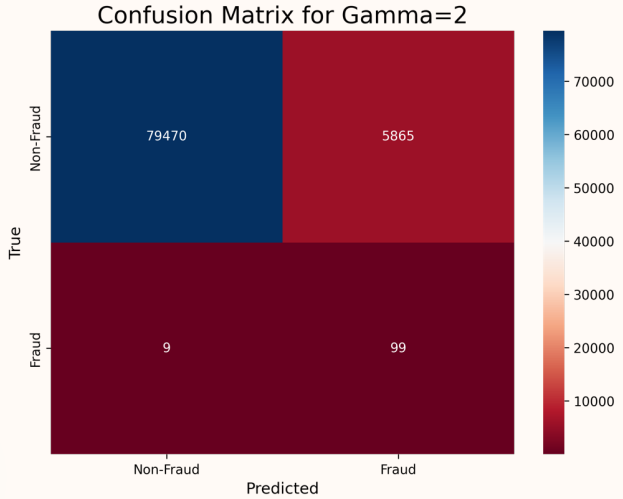
γ	Recall (Fraud)	Recall (Non-Fraud)	ROC AUC Score
0.5	0.935	0.851	0.893
1	0.880	0.951	0.915
2	**0.917**	**0.931**	**0.924**
3	0.880	0.971	0.925
4	0.889	0.950	0.920
5	0.926	0.898	0.912

Transformer Performance Under Focal Loss with Varying γ

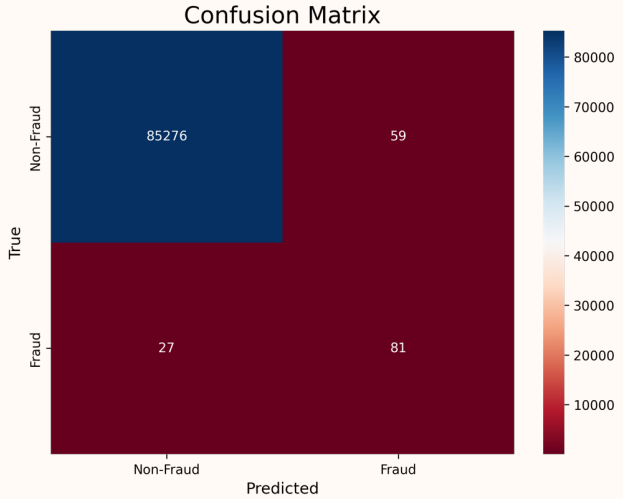
Experiments and Results

04. Transformer vs. LSTM (γ=2)

- **Transformer**: Fraud Recall = 0.917, ROC AUC = 0.92.
- **LSTM**: Fraud Recall = 0.75, ROC AUC = 0.875.
- **Insight**:
  - Transformer's attention mechanism better captures complex patterns and reweights features effectively, outperforming LSTM in both recall and overall performance.



Transformer With γ=2



LSTM With γ=2

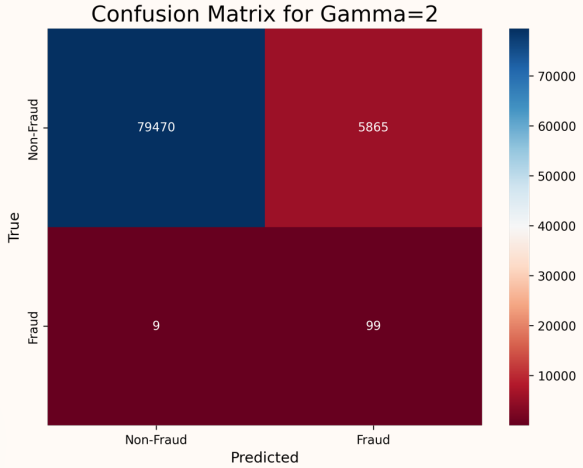
04

Discussion and Conclusion

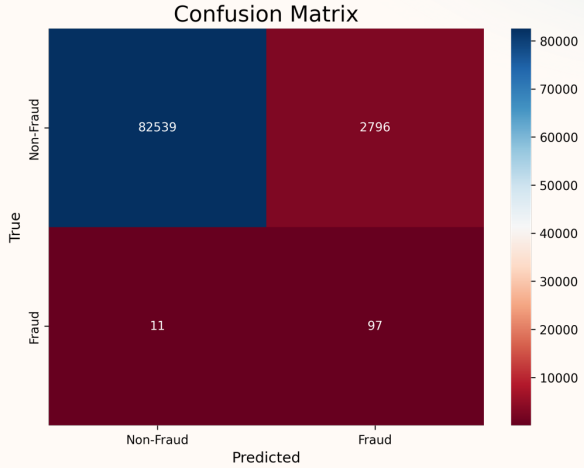
Discussion and Conclusion

01. Discussion: Transformer vs. Logistic Regression

- **Logistic Regression**:
  - Surprising effectiveness after SMOTE.
  - **Fraud Recall = 0.90, ROC AUC = 0.93**.
- **Transformer**:
  - **Fraud Recall = 0.917, ROC AUC = 0.92 (with γ=2)**.
  - More powerful at capturing nuanced patterns but requires careful tuning.



Transformer With γ=2



Logistic Regression With SMOTE

Discussion and Conclusion

02. Conclusion and Insights

- **Key Achievements**:
  - Demonstrated the potential of Transformer with Focal Loss to surpass Logistic Regression for fraud detection.
  - Balanced improvements in minority-class recall and overall performance.
- **Takeaways**:
  - Complex models require thoughtful parameter tuning and preprocessing.
  - Simpler models like Logistic Regression can still be competitive when data is properly balanced.

03. Limitations and Future Work

- **Limitations**:
  - **Validation Strategy**: The absence of a separate validation set may risk overfitting.
  - **Feature Scope**: Analysis limited to PCA-transformed and basic transaction-level features, without incorporating additional behavioral information.
- **Future Directions**:
  - Employ time series cross-validation to better evaluate model generalization.
  - Investigate model interpretability tools (e.g., SHAP) to assist fraud analysts in understanding predictions.

Thanks!