Henry Faya
4/15/2025

Fraud Detection with Neural Networks: Addressing Class Imbalance and Model Explainability

1. Dataset Selection and Motivation

Fraud Ecommerce Transaction Dataset:
9.36% fraudulent transactions versus 90.64% legitimate transactions.
Twitter Human/Bots Dataset:
33.19% bot accounts versus 66.81% human accounts.

These datasets were chosen to demonstrate how class imbalance affects model performance in different contexts and to explore how models can be improved through various techniques. The selection also reflects common fraud detection scenarios across different domains (financial transactions and social media), where identifying the minority class accurately is critical.

2. Model Architecture and Training Setup

Neural Network Architecture
For both datasets, I implemented a feed-forward neural network with the following architecture:
Input Layer → Dense(64, ReLU) → Dropout(0.4) → Dense(32, ReLU) → Dropout(0.4) → Dense(16, ReLU) → Dense(1, Sigmoid)
Key architectural features:

- L2 Regularization: Applied to all dense layers with a factor of 0.001 to prevent overfitting
- Dropout Layers: Set at 0.4 to reduce co-adaptation of neurons and improve generalization
- Binary Classification Output: Single sigmoid neuron for binary fraud/non-fraud prediction

Training Setup
The models were trained with the following configuration:

- Optimizer: Adam with default learning rate
- Loss Function: Binary cross-entropy
- Batch Size: 64
- Validation Split: 20% of training data, stratified to maintain class distribution
- Early Stopping: Based on validation AUC with patience of 10 epochs
- Preprocessing: Standardization for numerical features, one-hot encoding for categorical features

Class Imbalance Techniques
Four different approaches were implemented to address class imbalance:

1. Baseline: Standard training without any balancing technique
2. Random Oversampling: Duplicating minority class examples to balance class distribution
3. Random Undersampling: Reducing majority class examples to match minority class
4. Synthetic Minority Over-sampling (SMOTE): Creating synthetic examples of the minority class

5. Class Weights: Adjusting the loss function to penalize misclassification of minority class more heavily

3. Performance Metrics Before and After Balancing

Fraud Transaction Dataset Results

Model	AUC	Precision	Recall	F1-Score
Baseline (No balancing)	0.767	0.90	0.53	0.67
Random Oversampling	0.767	0.82	0.53	0.65
Random Undersampling	0.765	0.84	0.53	0.65
SMOTE	0.766	0.79	0.54	0.64
Class Weights	0.768	0.83	0.53	0.65

Twitter Bot Detection Dataset Results

Model	AUC	Precision	Recall	F1-Score
Baseline (No balancing)	0.835	0.65	0.71	0.68
Random Oversampling	0.843	0.59	0.86	0.70
Random Undersampling	0.836	0.57	0.87	0.69
SMOTE	0.842	0.59	0.85	0.69
Class Weights	0.839	0.58	0.86	0.69

Key Observations

1. Precision-Recall Tradeoff: A clear tradeoff is visible in both datasets. In the fraud dataset, the baseline model achieved the highest precision (0.90) but with relatively low recall (0.53). In the Twitter dataset, balancing techniques significantly improved recall (up to 0.87 with undersampling) at the cost of precision.
2. Consistent Recall Challenge in Fraud Dataset: Surprisingly, all methods produced very similar recall values around 0.53-0.54 for the fraud dataset, suggesting that approximately half of the fraudulent transactions remained undetected regardless of the balancing technique used.
3. Improved Recall in Twitter Dataset: In contrast to the fraud dataset, balancing techniques substantially improved recall for bot detection, increasing from 0.71 (baseline) to 0.86-0.87 (with balanced approaches).

4. LIME Analysis and Visualizations

Fraud Dataset Key Features

LIME analysis revealed that the most influential features for fraud prediction were:

1. Time between signup and purchase: Short duration between account creation and transaction was the strongest fraud indicator
2. Purchase hour: Late night purchases (between midnight and 5 AM) showed higher fraud probability
3. Source channel: Certain acquisition channels had higher fraud rates
4. Browser type: Specific browsers or unusual browser choices correlated with fraud
5. Age: Younger ages showed moderate correlation with fraudulent activity

For example, examining a misclassified instance (predicted as non-fraud but actually fraud), LIME showed the model failed to properly weigh the unusual time-of-day pattern combined with demographic features.

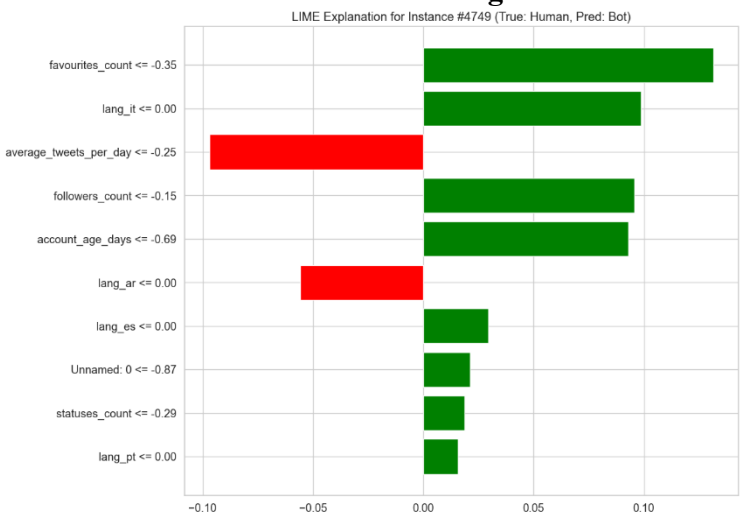
Twitter Bot Detection Key Features

For the Twitter bot detection model, LIME highlighted these top predictive features:

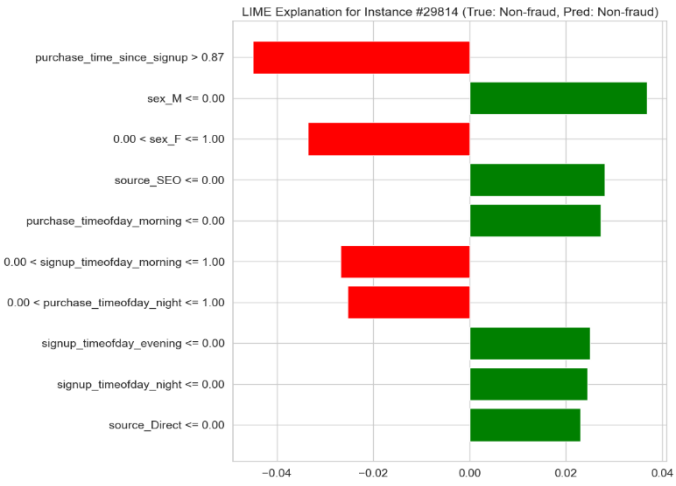
1. Average tweets per day: Extremely high posting frequency strongly indicated bot accounts
2. Account age: Newer accounts had higher likelihood of being bots
3. Default profile settings: Accounts with default settings were less likely to be bots
4. Friends to followers ratio: Unusual ratios indicated automated behavior
5. Geographical information: Human accounts more frequently enabled location services

Based on the LIME explanations of misclassified instances, we observed that some sophisticated bots with carefully managed activity patterns (moderate tweet frequency and balanced follower ratios) were able to evade detection.

Twitter LIME Image:



Fraud Ecommerce LIME Image:



5. Reflection on Class Imbalance and Explainability

Insights From Class Imbalance

Working with these two datasets with different imbalance levels revealed several important insights:

1. Imbalance Severity Affects Improvement Potential: The more severely imbalanced fraud dataset (9.36% fraud) showed less dramatic improvements from balancing techniques compared to the Twitter dataset (33.19% bots). This suggests that extreme imbalance poses fundamental detection challenges that simple resampling cannot fully address.
2. Balancing Techniques More Effective on Twitter Dataset: Random oversampling and SMOTE showed more significant improvements in the Twitter dataset (AUC increased from 0.835 to 0.843) compared to the fraud dataset where improvements were minimal.
3. Recall Improvements Varied by Dataset: While balancing techniques barely affected recall in the fraud dataset, they substantially improved recall in the Twitter dataset (from 0.71 to 0.87), suggesting that the effectiveness of balancing techniques may depend on the underlying data distribution and feature set.
4. High Precision vs. High Recall Trade-offs: The business implications of the precision-recall tradeoff became evident. For fraud detection, high precision might be preferred to avoid falsely accusing legitimate customers, while for bot detection, high recall might be more important to catch most automated accounts.

Insights from Explainability

The LIME analysis provided valuable insights that would not have been apparent from performance metrics alone:

1. Feature Importance Differences: Both datasets showed different key predictive features, with temporal features dominating fraud detection and activity patterns dominating bot detection.
2. Error Pattern Identification: LIME helped identify patterns in false negatives (missed fraud/bots), revealing limitations in the models' ability to detect sophisticated fraud that mimics legitimate behavior.
3. Feature Engineering Opportunities: The analysis highlighted potential new compound features that could improve detection, such as ratios between behavioral metrics or more sophisticated time-pattern analysis.
4. Contextual Explanations: LIME provided context-specific explanations for individual predictions, which is crucial for high-stakes domains where human reviewers need to understand why a transaction was flagged.
5. Trust Building: The ability to explain why specific cases were flagged enhances the usefulness of the model in practical applications where human analysts need to review potential fraud cases.