**Project Proposal -- Credit Card Fraud Detection**

**Abstract**

This project will study automatic fraud detection in consumer credit card transactions. Using two datasets (one synthetic and one real) we will build various supervised models and evaluate their performance. We will develop classification models using convolutional neural networks or xgboost and sequence-based models using recurrent neural networks with attention and compare their performance on the problem. We may also explore anomaly detection methods like one-class SVM or isolation forests.

**Introduction of Problem and Impact**

Credit card fraud detection is a crucial part of risk management and daily operations for every bank/transaction processor. It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. A valid and efficient detection of fraudulent activities can help both card holders and relevant business owners avoid financial loss, as well as lower operation cost of banks.

Data mining/collection is relatively straightforward in industry - when a customer reports or is alerted about a fraudulent transaction, a customer service representative can contact them and flag certain transactions as fraudulent before canceling their credit card. This along with the customer’s previous purchase history can be used as a supervision signal for detecting fraud. However, this data is very difficult to share publicly because it includes sensitive and/or personally identifiable information (namely the customer’s purchase history) and the problem is adversarial in nature (disclosure may give identity thieves a sense of how banks detect fraud etc). In addition, the data suffers from non-stationarity (changes from month-to-month) and sparsity (most transactions are non-fraudulent). Thus, being outside an industry setting we will have to use either synthetic data or anonymized data to study the problem.

The task of catching credit card fraud can be considered as a supervised learning problem. Charges show up on each customer's account, so fraudulent charges are usually caught - if not initially by the company, then later by the customer when account activity is reviewed. We can assume that nearly all fraud is identified and reliably labeled, since the legitimate customer and the person perpetrating the fraud are different people and have opposite goals. Thus credit card transactions have reliable labels (fraud and legitimate) that may serve as targets for a supervised technique.

**Proposed Solution**

In fraud detection domains, as we mentioned earlier, there are no universal numerical equations or criteria to define fraudulent activities. Features that help us with fraud detection need to be “patterns" that have a certain degree of generality.

One approach we could take is to do supervised classification where our input is a transaction plus any additional features involving the customer or their previous purchase history and our target variable is whether the transaction is fraudulent or not. For this we could use standard approaches like xgboost or convolutional neural networks (CNN’s).

A second approach we could take is sequence-based models, where instead of treating each transaction as iid we treat a customer’s transaction history as a single observation/time series (e.g. input) and either classify each transaction as fraudulent/not or try to predict if the entire time series was fraudulent or not (since a customer service representative can contact the customer either way). For the first output, we could try a conditional random field or a recurrent neural network and for the second we could try a recurrent neural network with attention over the whole time series.

Given the class imbalance present in our second dataset, we could also use anomaly detection methods. These methods generally posit the existence of some distribution from which customers’ transactions are drawn and try to find “outliers”. Two potential methods we could use are one-class SVM and isolation forests.

A potential challenge we face is the class imbalance in the second dataset. We could use a synthetic upsampling technique like SMOTE or something else to try to augment the data. We could also try downsampling, although the limited number of fraudulent examples in the dataset may make that difficult.

**Programming Languages**

We are using Python for this study, since it has the most relevant packages we would need and natural integration with deep learning frames like PyTorch.

**Performance Evaluation**

Given the significant class imbalance in our second dataset (and in the real-world dataset), accuracy is an inappropriate choice for a metric and AUC is also questionable. We will compute the precision/recall/F1 score for the fraudulent class and look at the precision/recall curve. Given the recall-oriented nature of the problem (it’s better to flag transactions than not), we will also look at various thresholds from our model.

**Data**

We are using two datasets from Kaggle for this study.

**Dataset 1: Synthetic Financial Datasets for Fraud Detection**

Content: This dataset simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.

Headers:

step - maps a unit of time in the real world.

type - CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

amount - amount of the transaction in local currency.

nameOrig - customer who started the transaction.

oldbalanceOrg - initial balance before the transaction.

newbalanceOrig - new balance after the transaction.

nameDest - customer who is the recipient of the transaction.

oldbalanceDest - initial balance recipient before the transaction.

newbalanceDest - new balance recipient after the transaction.

isFraud - This is the transactions made by the fraudulent agents inside the simulation.

isFlaggedFraud - The business model aims to control massive transfers from one account to another and flags illegal attempts.

**Dataset 2: Credit Card Fraud Detection**

Content: The datasets contain transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for $0.172\%$ of all transactions.

Headers: It contains only numeric input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, the dataset does not contain the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are `Time' and `Amount'. Feature `Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature `Amount' is the transaction Amount. Feature `Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

**Impact**

As outlined in the introduction, the business value of automatic fraud detection is relatively clear - customers lose money due to fraud and may choose a different credit card provider if we don’t detect fraud and there are simply too many daily transactions for business analysts to manually review these for fraud. Thus we must rely on some form of machine learning to flag potential fraud.