AMERICAN UNIVERSITY OF BEIRUT

MAROUN SEMAAN FACULTY OF ENGINEERING & ARCHITECTURE

Department of Electrical and Computer Engineering

EECE 490 – Introduction to Machine Learning

Real Time Fraud Detection Using Machine Learning

Project Proposal

Abdullah Itani

Abdellatif Itani

Mohammed Omar Mkhallati

Table of Contents

Motivation

Financial fraud and cybercrime are currently some of the most prevalent crimes in our digital day and age. There are always new and innovative ways used by criminals to manipulate their way into a person's bank account, whether it be through hacking, or through social engineering. This is a growing issue with cybercrime expecting to cost up to $10.5 trillion annually by 2025.

Banks and financial institutions face the issue of fraud and cybercrime multiple times, and it is essential to build a strong fraud detection and prevention system in order to counteract any possible malpractice.

Fraud does not only impact the customer, where they lose money that could add up to their entire life savings; it also affects banks heavily since the more people get defrauded for their money, the less they will trust the banks. This causes people to turn to other forms of financial institutions such as crypto, or as were seeing a lot in Lebanon, independent platforms that allow users to deposit and withdraw money with a certain fee (Whish, MyMonty, etc.).

Regardless of which financial institution one might use, there is always a chance of them getting defrauded for their money. There are multiple examples of crypto fraud as well as multiple Ponzi schemes.

Our goal through this project is to develop a system that will use AI and machine learning in order to properly label a transaction as fraud or not. We will use the classic available methods and improve upon them. We will also utilize the customers history of transactions in order to determine whether a transaction is fit with what is usually taking place or if it is an intruder.

Existing Solutions

There are a plethora of machine learning models and datasets that relate to the topic of fraud detection. The largest financial institutions in the world, PayPal, Visa and Mastercard along with the largest banks around the world, all use a similar fraud detection technique. The issue with the available fraud detection techniques is the false-positive rates, where some transactions are labeled as fraud when they are not.

We will mainly utilize Kaggle in order to find both datasets and previous models. Upon searching, we found a competition that was done by IEEE called the *IEEE-CIS Fraud Detection Challenge* and a whole forum dedicated to this topic. However, we realize that most models we found are not based on an individual's prior transactions but rather on large-scale datasets. This is where we aim to improve, our model will learn from each users' transactions and adjust the prediction model accordingly.

Our model will reduce the rate of false-positives greatly and will improve a user's experience overall as opposed to other available models. We will also aim to make our model more dynamic since with each passing day, there are more and more fraud techniques, and our model must be built robustly to reflect the changes that might occur.

Project Goal

We aim to create an efficient machine learning model that is capable of detecting fraudulent online transactions in real time. We will use deep neural networks and logistic regression to determine whether a transaction is legitimate or not. We want to implement and improve a user's experience thus we will cater our model to learn from the user's past transactions and history in order to reduce the false-positive rates and make it more personal per user. On top of that, we want our model to be able to adapt to any new types of fraud that might appear to ensure it is efficient and accurate.

Methodology

1. Data

We will be using datasets from Kaggle that are publicly available such as the IEEE-CIS Fraud Detection Dataset. This dataset includes examples of transaction data with timestamps, locations, amounts, as well as device types. We will also use our user's history to build a dataset of previous purchases, devices used, location, etc.

2. Programming Languages & Tools

The model will be implemented using python with the following libraries:

1. Pandas for data manipulation.
2. Scikit-learn for logistic regression.
3. TensorFlow for deep neural networks.
4. NumPy for computation.
5. Matplotlib for data plotting.

3. Machine Learning Models

Logistic Regression

We will have to use a typical binary classification where we must predict whether our output is detected as fraudulent (Binary 1) or legitimate (Binary 0).

Deep Neural Networks

Deep neural networks will be used for the definition of relationships between features in order to identify fraud techniques outside the scope of logistic regression.

4. Model Training & Validation

We will train the model based on previous transaction data and some synthetic data that will emulate real-life fraud situations. We will also go through the process of cross-validation and adjusting our hyperparameters in order to ensure the models accuracy.

5. Real-Time Adaptation

We will try and allow our system to retrain on a basis that will keep it up to date with any new fraud techniques. This means we update our data every certain time period to ensure the efficiency of detecting fraud.

Plan and Timeline

1. *Data Collection*

   We will work on gathering data from all public datasets available on the internet and process it in order to prepare it to be used in our training model.

2. *Feature Engineering*

   We will go through our data and extract any relevant features and create any new features where we see fit in order to improve the model.

3. *Model Development*

   We start with a logistic regression and train the model based on our data. We will also develop a deep neural network that will focus on the more complex fraud patterns.

4. *Real-Time Detection System*

   We want to build a system around our model that ensures that the model will be able to work in a seamless way and instantly.

5. *Testing & Validation*

   We need to test the model we developed and go through the tuning of hyperparameters in order to develop the best model.

6. *Optimization & Deployment*

   We will deploy the model and program at the end after we go through all the previous steps.