Group 23 | Project Report

One-Class Adversarial Nets for Fraud Detection

Members: Shai Siso
Shahar Avital
Golan Yacobov
Sean Friedman

Project environment: Python

Project libraries: Tensorflow, Keras, Numpy, Sklearn, Pandas

Project description: In this project, we develop one-class adversarial nets (OCAN) for fraud detection with only benign users as training data.

project study: We first study the RNN from Youtube (M.I.T lectures) and other sources after we learnt the basics of neural networks from the lectures. Then we got familiar with LSTM and autoencoders. The GAN research was the toughest job, so we learnt it for a while.

Project study: One-Class Adversarial Nets for Fraud Detection
https://arxiv.org/pdf/1803.01798.pdf

Dataset: UMDWikipedia dataset

X_v8_4_50_Ben.npy – dataset for benign users
X_v8_4_50_Van.npy - dataset for vandal users (for testing)
Dataset is located on the project folder in data/wiki

Project flow:

- Load dataset of benign and vandal users from UMDWikipedia datasets.
- Build the autoencoder with LSTM layers.
- Train the autoencoder with the benign users.
- Get the hidden representations of the benign and vandal users.
- Build a complementary GAN.
- Train the GAN with the benign users hidden representation as input.
- Test recognition of both benign and vandal users.
- Check the predict results.

Running lstm_autoencoder.py will train the first part of the algorithm and will save the users representation to file.
Ocan.py will train gan part and will show the results of the testing.
*We have also attached ready-made user representation files after lstm training so that the Ocan file can be run straight.
The bg_utils.py is for general methods.

Project results: As expected, the accuracy and precision were relatively high to other algorithms from the article.

Project conclusion: In this project, we have developed OCAN that consists of LSTM Autoencoder and complementary GAN for fraud detection when only benign users are observed during the training phase. During training, OCAN adopts LSTM-Autoencoder to learn benign user representations, and then uses the benign user representations to train a complementary GAN model.