Detection of anomalous behavior in Credit Card transactions with an One-Class Generative Adversarial Network
Agenda

➢ Deep Learning for Anomaly detection

➢ Introduction Dataset
  o Feature Engineering
  o Dimensionality Reduction

➢ Approach: One-Class Generative Adversarial Network (OOGAN)
  o Autoencoder
  o Generative Adversarial Network
  o OCGAN Framework

➢ Training Process
Research problem: Identification of suspicious transaction behavior

Major approaches: Misuse detection vs. Anomaly detection

Issues:
- Highly-imbalanced datasets
- Real-life transaction data

Approach with Deep Learning:
- Definition of a „normal region”
- Generate synthetic fraud samples for training the model
What are our data?

➢ Anonymized Credit card transactions from German and Austrian account holders which were obtained from a real-world application
➢ Highly-imbalanced: 248,612 transactions with 3252 fraud records (~1.3%)
➢ 42 primary attributes:
  o 26 non-numeric variables (21 categoricals & 5 datetime)
  o 6 numeric variables
➢ Problem: Feature may not be meaningful enough to depict the complex transaction behavior
What is the added value of Feature Engineering?

- Applying domain knowledge onto the data
- Support the model’s learning by providing derived attributes
- Decompose features to reduce categorical variance
- Mapping of transaction behavior
t-SNE: who’s your neighbor?

- Dimensionality Reduction for high-dimensional feature spaces
- Findings:
  - Frauds location in the outer regions
  - Significant diffusion of fraudulent and normal instances
- How to separate them?
Using an Autoencoder to separate classes

➢ Standard-Feedforward Neural Network $\rightarrow$ transformation of feature vectors into representations

➢ Learning representations of the latent structures in the given data

➢ Well-suited for the definition of an area in which only normal instances are located

Learned Representations by the Autoencoder

- Visualization of latent representations
- Encoded instances allow proper distinction
- **Solved:** Definition of a normal region – done!
Handling class imbalances with Generative Adversarial Network

- GAN’s architecture consists of two adversary Neural Networks
- Generator and Discriminator competing against each other in a minimax-game [Nash, 1951]
- Discriminator Network as a simple classifier | Generator Network creates synthetic samples

[see https://developers.google.com/machine-learning/gan/gan_structure]
OCGAN’s generated synthetic frauds
Design of the OCGAN-Framework

Latent representation of normal instances

Noise distribution $p_z$

random noise $z$

Generator $G$

$G(z)$

Generated distribution $p_G$

Real distribution $p_x$

Discriminator $D$

$\max_{D} \mathbb{E}_{x \sim p_{x}}[\log(D(x))] + \mathbb{E}_{z \sim p_{z}}[\log(1 - D(G(z)))]

\min_{G} \mathbb{E}_{z \sim p_{z}(z)}[\log(1 - D(G(z)))]$
Thanks for your attention!

Tim Schmidt

Position: Software Engineer

E-Mail: t.schmidt@finatix.de

Finatix GmbH

Barfußgäßchen 12 | 04109 Leipzig