

Welche Betrugsfälle sollen erkannt werden?

In dem Meeting konnten prinzipiell zwei verschiedene Betrugsarten herausgearbeitet werden:

- 1) Identity-Fraud
- 2) Transaction-Fraud

Es wird in der Praxis vielleicht eine Kombination beider nötig sein.

Identity-Fraud

Hierbei geht es um gefälschte Kundendaten.

...

Transaction-Fraud

Hierbei sollen auffällige Muster in den Transaktionen erkannt werden. Das Essentielle dieser Betrugserkennung sind die Daten und der Algorithmus. Für diese Punkte hatte Accenture Lösungsvorschläge gebracht.

Daten

Die Transaktionsdaten können random dazu generiert werden. Am besten wäre es aber, wenn wir diese im Internet finden könnten. zB. Durch die freien Datensätze unten.

Betrugsfälle:

Diese müssen selbst dazu modelliert werden.

Beispiel der AG war:

- einen Geldwäsche-Ring zu simulieren
- Schritt für Schritt auffälligeren Betrugsfälle einzubauen, um Betrugserkennung zu testen.

Es sollte versucht werden über Betrugsmethoden zu recherchieren, um so möglichst wirklichkeitsnahe Betrugsfälle einzubauen.

Algorithmus

Es handelt sich um einen Machine Learning Algorithmus. Dieser soll unter der Verwendung von Neo4j APOC implementiert werden. APOC stellt eine Reihe von Algorithmen bereit, die nur mehr mit Daten „gefüttert“ werden müssen.

GUI

Die GUI muss nicht aufwendig sein. Sie kann wie der Neo4j Browser implementiert werden. Die verschiedenen Quirles werden durch Buttons ausgeführt, statt dass man sie selbst eingeben muss. Weiteres soll es noch ein paar Eingabefelder besitzen.

Zusätzlich soll es noch eine Short-List geben. Dies ist eine Box, die die Transaktionen/Kunden anzeigt, die besonders auffällig sind (probability fraudulent customers). Auf diese sollte man drücken können, um so dessen Graph angezeigt bekommen (Manuel Traversieren).

Zsm. Recherche

- Betrugsmuster
- Transaktionsdaten
- APOC

Neo4j APOC

https://neo4j-contrib.github.io/neo4j-apoc-procedures/#_graph_algorithms

Freie Datensätze

<https://snap.stanford.edu/data/>