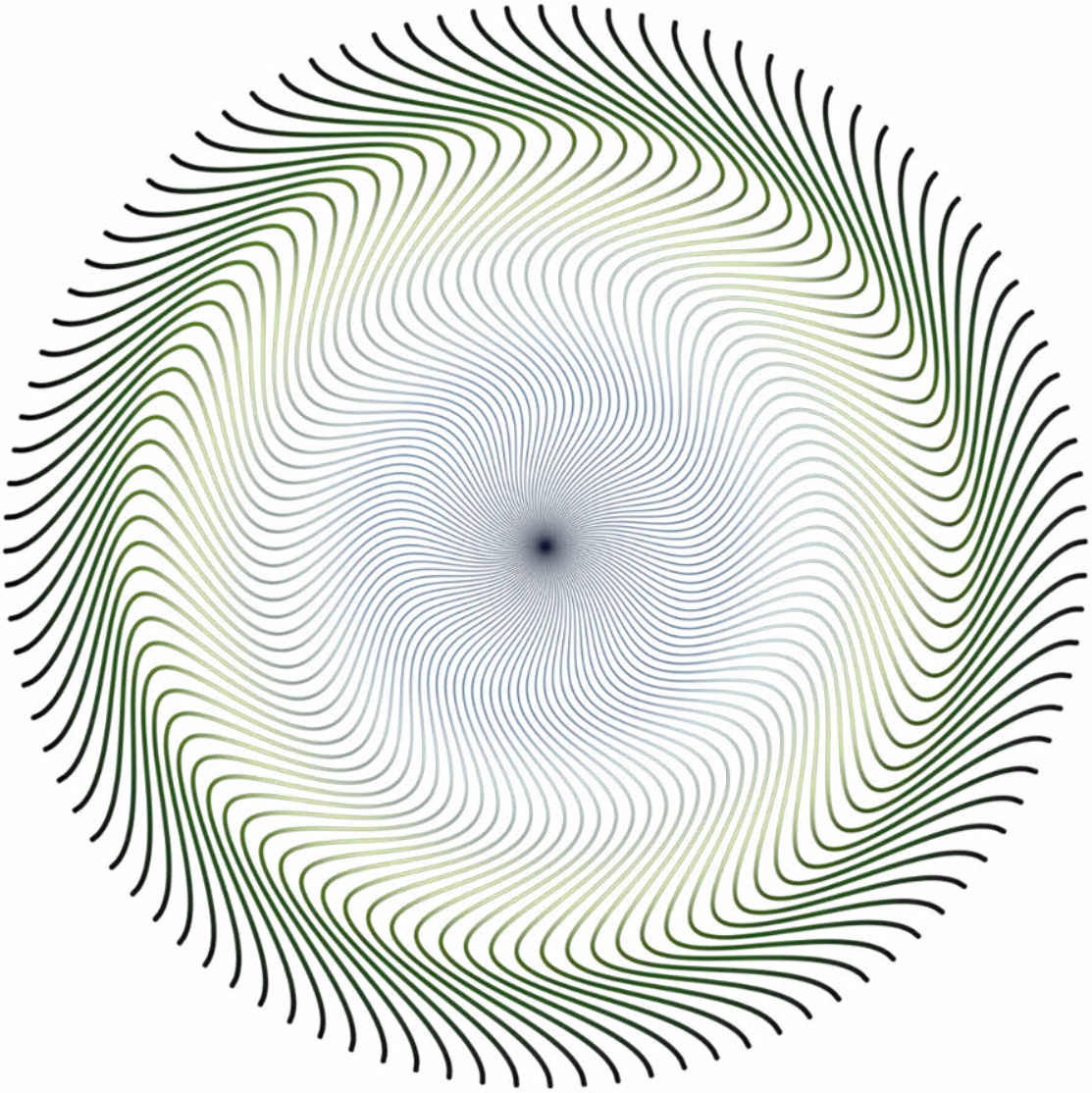
Deloitte.

neo4j



Betrugserkennung mithilfe von Graph-Technologie

Vorwort

Betrügerische Anträge für COVID-19-Überbrückungshilfen haben ein hohes Schadenspotenzial. Dieses Whitepaper beschreibt exemplarisch ein Projekt, dessen Inhalt das Erkennen und Bearbeiten fraudulenter Anträge gewesen ist. Im Folgenden wird die Vorgehensweise zur Erstellung des Datenmodells, sowie die Speicherung der Daten dargestellt. Hierzu zählt sowohl die Auswahl einer geeigneten Lösung, als auch die Erstellung eines Proof of Concept (PoC) mittels Graph-Technologie. Der hier vorgestellte Ansatz zeichnet sich zudem durch seine Wiederverwendbarkeit aus.

MAKING AN IMPACT THAT MATTERS since 1845

Das Projekt wurde in Zusammenarbeit von Deloitte und Neo4j durchgeführt. Neo4j ist der führende Anbieter der sogenannten Graph-Technologie mit Sitz in Malmö, Schweden und San Mateo, Kalifornien. Deloitte ist ein Wirtschaftsprüfungs- und Beratungsunternehmen, das unter anderem Unterstützung in den Bereichen Risiko- und Finanzberatung sowie Leistungen bei Wirtschaftsprüfung und Steuerberatung erbringt. Für das vorliegende Projekt war der Bereich Financial Advisory – Forensic FSI verantwortlich, der sich mit der Prävention und Aufklärung von Wirtschaftskriminalität in der Finanzindustrie beschäftigt.

Hinweis: Bei den hier dargestellten Antragsnummern und Kennzahlen handelt es sich um fiktive Werte, welche lediglich der Veranschaulichung dienen.

Einleitung

Die COVID-19-Pandemie und die Maßnahmen zu ihrer Bekämpfung stellen viele Unternehmen und Selbstständige vor große Herausforderungen. Die Bundesregierung bot deshalb finanzielle Unterstützung an, die es Unternehmen und Selbstständigen ermöglichen sollte, selbst bei Schließungsmaßnahmen und Umsatzausfällen die laufenden Kosten decken zu können. Diese sogenannten Corona-Soforthilfen wurden auf Antrag bewilligt und ausgezahlt.

Die Prüfung der Anträge auf Corona-Soforthilfen wurde zum Großteil auf die Landesförderbanken übertragen. Um diese Aufgabe bewältigen zu können, war es für die Landesförderbanken notwendig, neue Arbeitsabläufe zu definieren.

Im Rahmen der Corona-Soforthilfen bestand die Schwierigkeit bei der Bearbeitung der eingereichten Anträge darin, dass zum einen auf eine schnelle Abarbeitung und zum anderen auf eine gründliche Überprüfung Wert gelegt werden musste. Ziel war es, die betroffenen Antragsteller (juristische und natürliche Personen) schnell auszubezahlen, damit diese nicht in Liquiditätsschwierigkeiten geraten und gleichzeitig die Antragsteller zu identifizieren, die unrechtmäßig an Fördermittel zu gelangen versuchten.

Folglich war eine intuitive Filtermethode nötig, mit der die Prüfung beschleunigt werden konnte. In einem ersten Schritt sollten dadurch auffällige Anträge herausgefiltert werden. Diese konnten dann einer eingehenderen Prüfung unterzogen werden, während unauffällige Anträge zu einer schnellen Auszahlung weitergeleitet wurden. Insbesondere vor diesem Hintergrund war es vielfach notwendig, die Antragsberechtigung auch nachträglich noch einmal zu überprüfen. Für die Umsetzung dieses Projektes (auch mit einem möglichen Fokus auf eine nachträgliche Überprüfbarkeit) war also einerseits ein Katalog nachvollziehbarer Kriterien erforderlich, die automatisch auf Anträge angewandt werden konnten. Andererseits bedurfte es einer anwenderfreundlichen Darstellung, um die Sachbearbeiter bei der weitergehenden Prüfung auffälliger Anträge zu unterstützen.

Technologien – Anforderungen des Projekts

Die Corona-Soforthilfen des Bundes sowie der Länder bestehen aus mehreren Förderprogrammen. Im Rahmen des ersten Förderprogramms kam ein lokales Verfahren für das Prüfen und Bearbeiten von Anträgen zum Einsatz. Bei den danach folgenden Förderprogrammen wurde anschließend von der Bundesregierung ein bundesweit einheitliches Verfahren entwickelt und angewendet. Es bestanden also verschiedene Datenquellen, die gemeinsam ausgewertet werden sollten. Die Lösung musste in der Lage sein, beide vorgenannten Quellen abzubilden, ohne dass Informationen dabei verloren gingen.

Ursprünglich wurde beschlossen, die Anträge isoliert voneinander zu betrachten und über vordefinierte potentielle Betrugsfaktoren zu bewerten, was sich jedoch als lückenhaft erwies. Übergreifende Auffälligkeiten konnten dadurch nicht erkannt werden, etwa das Stellen mehrerer Anträge durch eine Person oder aber das Eingehen überdurchschnittlich vieler Anträge von einer identischen Adresse.

Bei der Bearbeitung selbst war aus dem zugeteilten Antrag häufig nicht ersichtlich, weshalb dieser im Einzelnen in die Betrugsfallprüfung weitergeleitet wurde. Wenn ein Antrag beispielsweise manuell aufgrund eines zuvor auffällig gewordenen Namens in die Prüfung aufgenommen wurde, so wurde dieses Kontextwissen nicht mit übermittelt, was die weitere Bearbeitung deutlich erschwerte.

Die weitere Zuteilung und Bearbeitung der Anträge erfolgte über Excel Spreadsheets, welche jeweils immer nur einen kleinen Ausschnitt der vorhandenen Daten umfasste. Diese Listen wurden über einen SharePoint administriert und konnten nur durch eine Person gleichzeitig bearbeitet werden. Die manuelle Bearbeitung der Listen führte häufig zu Problemen beim Aktualisieren der Inhalte. Außerdem war die Bearbeitung durch das Hosting und durch den Umfang der jeweiligen Daten zeitlich sehr aufwendig.

Zum Anforderungsprofil der seinerzeit zu erarbeitenden Lösung gehörte dementsprechend, dass sie Informationen aus verschiedenen Datenquellen verarbeiten und vernetzt zusammenführen konnte. Außerdem sollten antragsübergreifende Auffälligkeiten identifiziert sowie für eine weitere Prüfung visuell und verständlich aufbereitet werden. Schließlich sollte die Lösung eine stabile Umgebung bereitstellen, in welcher mehrere Personen simultan und schnell Änderungen durchführen können.

Beschreibung der Herangehensweise

Wie wurden diese Anforderungen umgesetzt? Im Folgenden möchten wir kurz die wesentlichen Schritte unserer Herangehensweise in diesem Projekt erläutern und auf die wichtigsten Punkte hinweisen.

Schritt 1: Technologischen Anforderungen gerecht werden

In einem ersten Schritt galt es, eine passende Technologie zu finden, welche die oben genannten Kriterien erfüllt. Durch eine bereits länger bestehende Kooperation zwischen Deloitte und Neo4j lagen positive Erfahrungen mit sogenannten Graph-Datenbanken vor.

Die Neo4j-Graph-Datenbank speichert Daten in einem sogenannten Labeled-Property-Graphen (LPG). Graph-Datenbanken bestehen aus Knoten („Nodes"), die bestimmte Instanzen repräsentieren, sowie aus Relationen („Kanten"), mit denen diese verbunden sind. Knoten und Relationen werden auch in dieser Form auf Festplatte gespeichert. Erstere bekommen dabei Labels wie z.B. „Antrag" oder „Unternehmen". Relationen werden durch sogenannte „Typen" gekennzeichnet, die die Beziehungen darstellen, wie zum Beispiel „GESTELLT_VON". Ein einfacher Graph könnte also folgendermaßen aussehen:

(Antrag) – GESTELLT_VON->(Unternehmen)

Den einzelnen Knoten und Relationen können dann noch Eigenschaften („Properties") zugeordnet werden, damit auch weitere Daten wie Unternehmensname, Antrags-ID o.Ä. gespeichert werden können. Ein komplettes Abbild des Graphen wird als „Datenmodell" bezeichnet.

Schritt 2: Beschreibung der gesuchten Fragen und Antworten

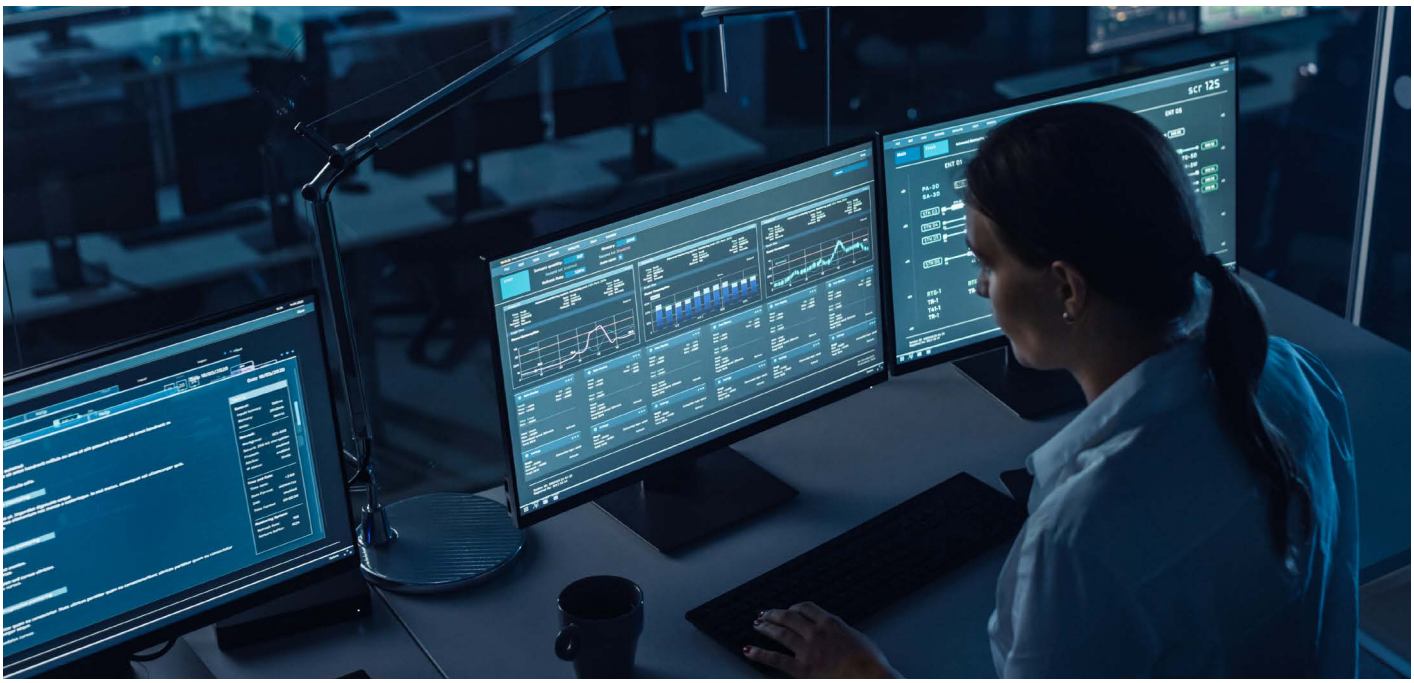
In einem nächsten Schritt wurde die Umsetzung des Projektes geplant. Insbesondere mussten die bereits zuvor definierten potentiellen Betrugskriterien für isolierte Anträge um eine übergeordnete Analyse auf der Beziehungsebene erweitert werden. Der bestehende sogenannte Betrugs-Wert stellte dabei eine Gewichtung der Kriterien dar. Hierfür waren nun weitere Kriterien zu definieren, um entsprechende Abfragen formulieren und die Ergebnisse optimieren zu können.

Hierbei wurde unterschieden zwischen sogenannten „weichen Kriterien" (Kriterienliste A), welche zwar ein Indiz für einen möglichen Betrugsverdacht darstellen können, jedoch alleinstehend nicht ausreichend für eine umfassende Betrugsprüfung sind und eine solche Prüfung nur bei Vorliegen mehrerer Kriterien durchgeführt wird, und „harten Kriterien" (Kriterienliste B), welche eine Betrugsprüfung zwingend auslösen.

Basierend auf der vollständigen Liste der Fragen und den zugehörigen Antworten wird dann entschieden, welche Anträge einer Betrugsverdachtsprüfung unterzogen werden sollten. Dabei galt je mehr Kriterien vorliegen, desto wahrscheinlicher ist der Antrag fraudulent. Entsprechend wurde dieser dann an verschiedene Sachbearbeiterteams weitergeleitet.

Zusätzlich zu diesen Kriterien wurden Kriterien ergänzt, die auf den Beziehungen zwischen verschiedenen Anträgen basieren, wie zum Beispiel:

- Hat eine natürliche Person in der Funktion als Selbstständiger oder als vertretungsberechtigte Person eines Unternehmens mehrere Anträge eingereicht?
- Wurden weitere Anträge der selbstständigen Person oder des Unternehmens in der Vergangenheit als auffällig eingestuft?



Schritt 3: Ableitung des Datenmodells
Aus den gesammelten Kriterien wurde in Zusammenarbeit mit den Experten von Neo4j ein Datenmodell entwickelt. Das ursprüngliche Modell wurde dabei in mehreren Iterationszyklen angepasst und optimiert, um einzelne Fragenkriterien besser beantworten zu können oder auch weitere Fragen mit dem Modell zu unterstützen. Das überarbeitete Datenmodell für unser COVID-19-Betrugsprojekt ist in Abb. 1 dargestellt.

In der grafischen Darstellung des Datenmodells lassen sich die herausgearbeiteten Entitäten (Nodes/Knoten) wie Antrag, Unternehmen, Person, IBAN-Sperrliste, Prüfung, Kriterienliste A (für weiche Kriterien) usw. entnehmen. Da die Informationen als separate Knoten gespeichert werden, können wiederholende Informationen förderprogrammübergreifend konsolidiert werden, was die weitergehende Analyse vereinfacht. Zu solchen sich wiederholenden Informationen gehören z.B. Adressen, die häufiger verwendet wurden, oder Unternehmen, die mehrere Anträge gestellt haben.

Die einzelnen Entitäten werden dann miteinander in Beziehung gesetzt, dargestellt durch die Pfeile (Relationen) zwischen den einzelnen Knoten. Dies wird in dieser Form in der Graph-Datenbank gespeichert und abgefragt. Dadurch können beispielsweise auch verschiedene Anträge einem Unternehmen zugeschrieben werden.

Was das Schaubild nicht zeigt, sind die sogenannten „Properties" der einzelnen Knoten und Relationen im Graphen. Diese Properties stellen gewissermaßen Datenfelder der einzelnen Knoten dar. In ihnen können Daten zum Antrag gespeichert werden, wie etwa die Antrags-ID, beantragte oder ausgezahlte Beträge und weitere wichtige Informationen. Auch diese können im Weiteren für Analysen und Visualisierungen verwendet werden.

Schritt 4: Daten aufbereiten und laden
Nach der erfolgreichen Erarbeitung eines ersten Datenmodells können Daten in die Graph-Datenbank geladen werden. Im Rahmen des Proof of Concepts wurde dieser Schritt zunächst testweise über verschiedene Excel-Tabellen vorgenommen. Zur Wahrung des Datenschutzes kamen hier nur pseudonymisierte Daten zum Einsatz. Diese wurden vorab durch eine Deloitte-eigene Softwarelösung generiert.

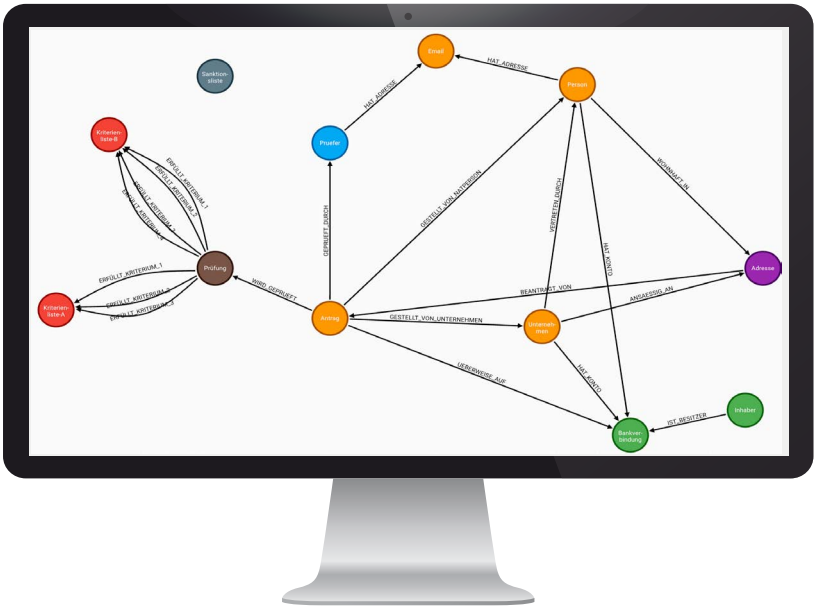
Insgesamt lässt sich der Datenimport in diesem Stadium noch weiter optimieren. Die Neo4j-Abfragesprache Cypher, die als das SQL der Graph-Datenbankwelt bezeichnet werden kann, stellt hierfür unterschiedliche Funktionen bereit. Beispielsweise können Formate wie CSV-Dateien, aber auch JSON- oder XML-Daten eingelesen werden. Auch der Zugriff auf Datenbanken mittels Standardschnittstellen wie JDBC oder ODBC ist möglich. Wenn ein Kunde schon ein sogenanntes ETL-Tool (ETL = Extract, Transform, Load) benutzt, kann auch dieses eingebunden werden, um den Graphen zu laden.

Im Kontext der Corona-Soforthilfen stieg die Anzahl an Förderanträgen täglich. Somit wurden permanent neue Datensätze generiert. Hier würde sich ebenfalls eine Schnittstellenlösung anbieten, über die ein Live-Zugriff auf die aufgebaute Datenbank gewährleistet wird.

Schritt 5: Datenqualitätsmanagement und Fuzzy Logic
Beim Laden der pseudonymisierten Daten wurden in einem Qualitätscheck Probleme durch abweichende Schreibweisen festgestellt. Beim Abgleich verschiedener Anträge bezüglich Adressen, Unternehmensnamen und natürlichen Personen lagen häufig unterschiedliche Schreibweisen in verschiedenen Anträgen vor. Das erschwerte die Identifikation von möglicherweise fraudulenten Anträgen.

Um dieses Problem anzugehen, wurden Textfelder wie Unternehmens- und Personennamen mittels Volltextindex indiziert. Dieser Mechanismus untersucht Texte im Stil einer Suchmaschine und bietet für die Suche dann „Fuzzy Logic" an. Das bedeutet, dass mittels Operatoren auch ähnliche Namen gesucht werden können. Je näher das Resultat am Suchausdruck ist, desto prominenter erscheint es in der Ergebnisliste. Ein spezieller Score wird berechnet und zeigt zusätzlich an, wie stark Ausdrücke einander gleichen. Hierbei handelt es sich um einen Schritt zur sogenannten Entity Resolution, wobei es um die Verbesserung der Datenqualität durch Identifizierung von Duplikaten und zusammengehörigen Datensätzen geht.

Abb. 1 – Datenmodell für COVID-19 Fraud Detection (ohne Properties)



Schritt 6: Fragen mittels Abfragesprache oder Visualisierungstools beantworten

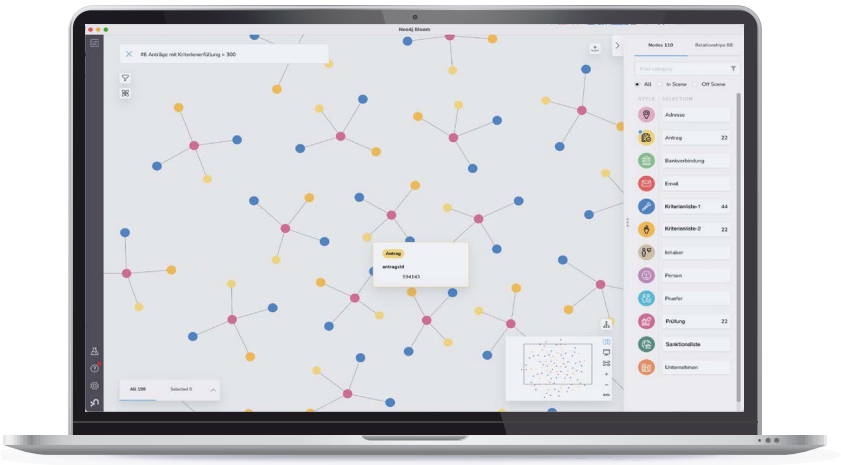
Sobald die Daten geladen und mittels der Entity-Resolution-Maßnahmen geprüft wurden, können sie über verschiedene Graphen abgefragt werden. Damit kann nun überprüft werden, ob die Liste der Fragen (s. Schritt 2) beantwortet und eine Bewertung durchgeführt werden kann. Hierfür kommen im Allgemeinen die Abfragesprache Cypher und weitere Visualisierungstools zum Einsatz. Im vorliegenden Projekt haben sich hierfür die Tools NeoDash und Neo4j Bloom angeboten.

Mit Hilfe von Neo4j Bloom kann in einem ersten Schritt überprüft werden, welche Anträge welche vorgegebenen Kriterien erfüllt haben. Dadurch werden die Originaldaten verifiziert, um auch die Qualität des aufgesetzten Systems sicherzustellen. Da durch dieses Verfahren weitere potentielle Betrugskriterien ergänzt wurden, können bei Suchanfragen auch mehr potentiell fraudulente Anträge erkannt werden, als dies beim Originaldatensatz der Fall ist. Dennoch kann festgestellt werden, ob die ermittelten potentiellen Betrugsfälle hier kongruent mit den originären Anträgen sind. Dafür lassen sich die durch dieses Verfahren erzielten Ergebnisse als Excel-Datei herunterladen und anschließend mit den Originaldaten abgleichen.

Die Neo4j-Bloom-Ansicht bietet einen visuellen Completeness Check. Außerdem kann durch die Oberfläche auch die Antragsbearbeitung durch visuell nachvollziehbare Kriterien gekennzeichnet werden. So lässt sich erkennen, weshalb ein Antrag in der Betrugsfallprüfung gelandet ist und worauf bei einer entsprechenden Prüfung der Fokus gelegt werden sollte.

Über diese Ansicht lassen sich aber nicht nur Anfragen zur übergreifenden Identifikation von potentiell fraudulenten Anträgen darstellen. Neo4j Bloom kann auch in der direkten Sachbearbeitung genutzt werden, um nach einzelnen Anträgen zu filtern und alle potentiellen Betrugsfaktoren sowie Properties und verbundene Knoten anzuzeigen. Zudem ist das simultane Bearbeiten einzelner Properties von Anträgen möglich, ohne dass es zu zeitlichen Verzögerungen kommt.

Abb. 2 – Neo4j Bloom – Bewertung von fraudulenten Daten analysieren

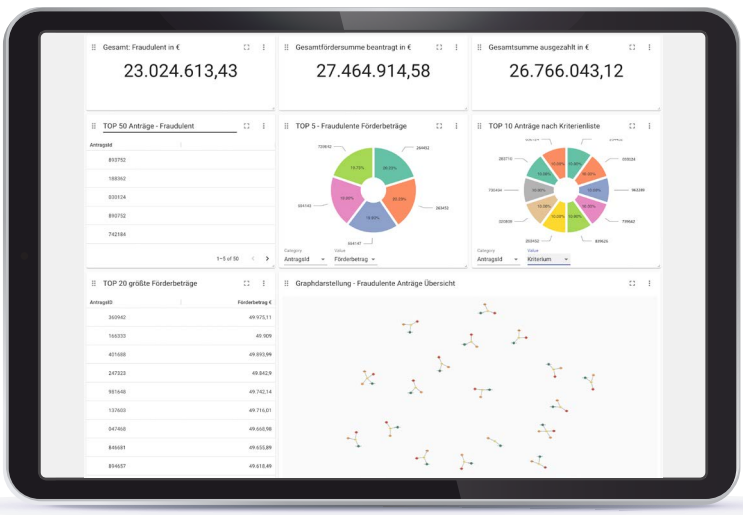


Die Vorteile der Nutzung von Neo4j Bloom liegen in der einfachen Suche und der visuellen Analyse der Datenmengen, die im Graphen (der Datenbank) gespeichert sind. Dadurch können Sachbearbeiter das Tool nutzen, ohne sich mit dem Datenmodell oder den gespeicherten Datenfeldern auskennen zu müssen.

Neben dem kann Neo4j Bloom um Abfragen in natürlicher Sprache erweitert werden (NLP, Natural Language Processing). Dies ermöglicht es Sachbearbeitern, bei der Datenanalyse Textsätze zu verwenden, beispielsweise: „Zeig mir alle Anträge von Unternehmen X mit einem Betrugswert größer Y." Bei diesen NLP-Abfragen können zusätzlich auch dynamisch generierte Suchwerte in der Textabfrage verwendet werden (Beispiel: X als Fuzzy-Suche eines Unternehmensnamens, Y als Betrugswert). Das erhöht die Flexibilität in der Nutzung.

Als weiteres Tool wurde im Rahmen der erarbeiteten Lösung NeoDash eingesetzt, um das Reporting von Anträgen zu optimieren, die aktuell in Bearbeitung sind. Bei NeoDash handelt es sich um eine Datenvisualisierungssoftware für Neo4j, mit der individuelle Dashboards erstellbar sind. Diese geben dann z.B. einen Überblick über den aktuellen Projektfortschritt oder noch ausstehende Anträge, Summen usw. NeoDash nutzt hierfür Echtzeitdaten aus Neo4j. Abbildung 3 liefert eine beispielhafte Darstellung eines solchen Reports.

Abb. 3 – Neodash – Dashboard der Kennzahlen



Weitere Schritte: Überarbeitung, Ausbau und Weiterentwicklung

Im Rahmen des beschrieben PoC stehen im weiteren Verlauf zusätzliche Anpassungen an: etwa die Überarbeitung des Datenmodells und der Abfragen, falls weitere Anforderungen hinzugekommen sind, oder weitere Verbesserungen der Abfragen nötig sind. Diese weiteren Optimierungsschritte werden zu einem späteren Zeitpunkt näher dargestellt.

Ferner ist es beabsichtigt, im Rahmen der Weiterentwicklung (in anderen Projekten) die Automatisierung der Prozesse weiter auszubauen und zu optimiert. Dies betrifft sowohl das Laden der Daten mittels ETL-Pipeline (ETL = Extract, Transform, Load) und eines kundenseitig eingesetzten Tools als auch das automatisierte Bewerten neuer Daten. Durch Letzteres kann sogar eine Früherkennung realisiert werden. So wird der gesamte Arbeitsprozess für die Bearbeitung wesentlich erleichtert und eine Gleichbehandlung bei der Bearbeitung der Anträge sichergestellt.

Je nach Kundenumgebung und Betriebsweise einer Lösung sind darüber hinaus weitere Schritte notwendig – etwa für die Inbetriebnahme, die operationelle Betreuung und das Monitoring der Lösung. Diese Punkte erfordern eine kundenspezifisch individuelle Umsetzung, weshalb sie hier ebenfalls nicht weiter erörtert werden können.

Übertragbarkeit des Ansatzes

Die Vorgehensweise beim Aufbau von Graph-Datenbanken ist grundsätzlich ähnlich. Die hier vorgestellte Lösung liefert ein gutes Beispiel für potentielle Betrugs Use Cases und kann in Teilen für andere Projekte wiederverwendet werden. Das Erstellen eines Datenmodells ist einfach und intuitiv umzusetzen, wenn das entsprechende Fachwissen (Domänenwissen) vorhanden ist. Von der Analyse des Problems bis zur ersten Abfrage vergehen bei solchen Projekten oft nur wenige Tage.

Für die potentielle Betrugserkennung bei Coronasofothilfe-Anträgen wurde im vorliegenden Projekt eine Graph-Datenbank erstellt, die auch als „Knowledge Graph“ (KG) bezeichnet wird. Der KG enthält dann die Wissensbasis für einen oder auch mehrere Use Cases.

Wenn der Graph aufgebaut ist, gilt es, diesen bedarfsorientiert weiterzuentwickeln und so die Business-Prozesse des Kunden noch umfangreicher zu unterstützen. Zu den weiteren Anwendungsgebieten neben der Prävention potentiell fraudulenter Transaktionen gehören z.B. andere Bereiche des Risikomanagements. Der Graph kommt als Wissensbasis für die Data-Science-Abteilung infrage, die den Ansatz mit ihren Methoden noch verbessern kann.

Zusammenfassung

Der Proof of Concept zur Erkennung von potentiellem Betrugssachverhalten bei Corona-Soforthilfen zeigt, wo die Stärken der Graph-Technologie liegen.

Besonders in Anwendungsfällen mit hoher Komplexität oder einer großen Menge an verknüpften Daten ist eine Bearbeitung mit Mitteln wie Excel oder einfach relationalen Datenbanken oft nicht zielführend. Hier bietet der Graph-Datenbank-Ansatz – wie dargestellt – eine zukunftsweisende Lösung.

Kontakte

Matthias Rode
Partner | Financial Advisory FSI
Tel: +49 151 58002270
mattrode@deloitte.de

Dr. Christoph Wronka
Director | Financial Advisory FSI
Tel: +49 69 75695 6037
cwronka@deloitte.de

Janina Uspelkat
Senior Consultant | Financial Advisory FSI
Tel: +49 40 32080 4908
juspelkat@deloitte.de

Deloitte.

Stand 10/2022