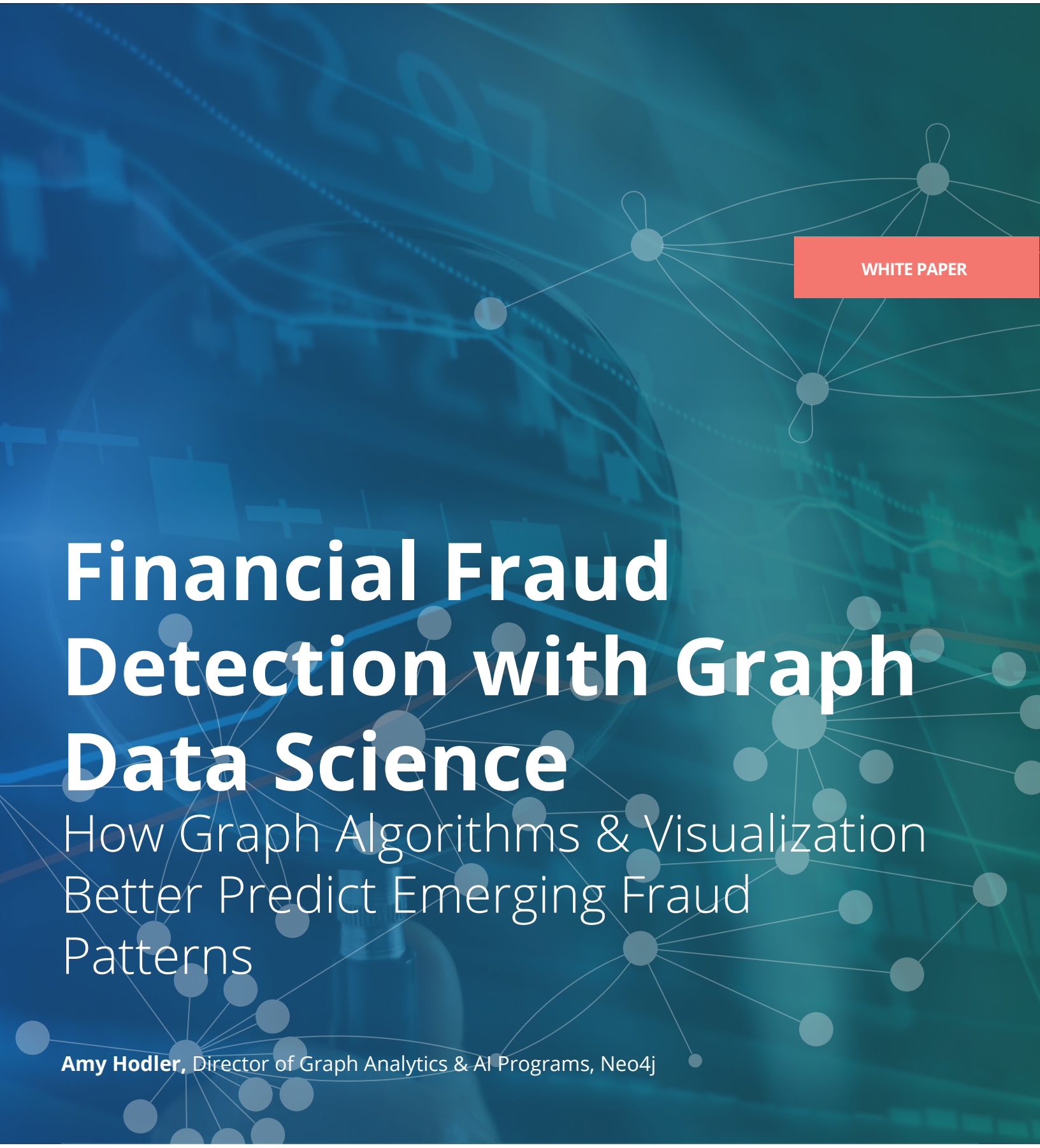


WHITE PAPER

Financial Fraud Detection with Graph Data Science

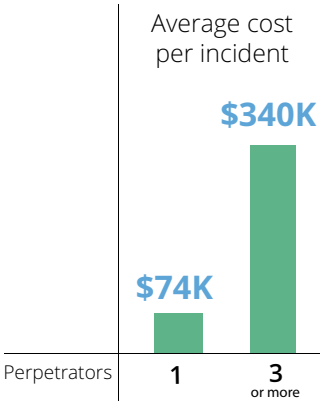
How Graph Algorithms & Visualization Better Predict Emerging Fraud Patterns

Amy Hodler, Director of Graph Analytics & AI Programs, Neo4j

White Paper

TABLE OF CONTENTS

FRAUD RINGS INCREASE LOSS



Financial Fraud Detection with Graph Data Science

Amy Hodler, Director of Graph Analytics & AI Programs, Neo4j

Introduction

Financial fraud is growing and it is a costly problem, estimated at 6% of the Global Domestic Product, more than $5 trillion in 2019.

Despite using increasingly sophisticated fraud detection tools – often tapping into AI and machine learning – businesses lose more and more money to fraudulent schemes every year. Graph data science helps turn this pattern around.

By augmenting existing analytics and machine learning pipelines, a graph data science approach increases the accuracy and viability of existing fraud detection methods. The end result: Fewer fraudulent transactions and safer revenue streams.

In this white paper, we'll take a closer look at how your data science and fraud investigation teams can tap into the power of graph technology for higher quality predictions in detecting first-party fraud as well as sophisticated fraud rings.

The Challenge of Detecting Financial Fraud

Stemming the wave of financial loss requires constant vigilance since fraud perpetrators continue to evolve their tactics, allowing them to evade detection.

Take for example one of the fastest-growing types of fraud in the U.S.: synthetic identity theft. Fraudsters meld various false and authentic elements (such as addresses, phone numbers, emails, employers and more) into a synthetic identity, which they then use for fraudulent purposes. Synthetic identities pass as real identities all too frequently. Traditional fraud models that consistently flag other types of high-risk identities miss 85% of synthetic identities according to ID Analytics.

At the same time, fraud rings – both small and large – are on the rise. With multiple parties involved in fraud, the associated loss skyrockets. In its 2018 Report to the Nations, the Association of Certified Fraud Examiners (ACFE) found a direct correlation between the number of participants and the cost of a fraud incident, rising from an average of $74,000 for one perpetrator to $339,000 for three or more perpetrators. Like pack hunters, fraudsters are a greater threat when they work together.

The question becomes how to reduce losses from fraud given these challenges.

Why Current Tactics Fail to Identify All Fraud

Data scientists have developed rigorous machine learning (ML) and analytics models to detect fraud. However, most data science models omit something critically important: network structure.

Research on social network analysis highlights the predictive power of analyzing network structure. As James Fowler says in his book, *Connected*, "Increasingly we're learning that you can make better predictions about people by getting all the information from their friends and their friends' friends than you can from the information you have about the person themselves."

Network analysis captures the inherent relationships between data elements. We are accustomed to thinking of social network data as a graph, but in fact, *any* type of data can be represented in this way. For example, it's possible to visualize account holders and their information as a graph.

When you analyze the network structure of account holder information, you may see (as shown in the figure below) that multiple account holders have the same phone number or the same identification number. Sharing the same elements may indicate synthetic identity fraud. These types of fraud signals are difficult to uncover without an effective way to examine the vast network structure of thousands – or even millions – of account holders.

Tabular data models, with data organized in rows and columns, are not designed for capturing the complex relationships and network structure inherent in your data.

Analyzing data as a graph enables you to reveal and use its structure for predictions, and with a graph database you can persist these connections for later analysis.

Enter Graph Data Science for Fraud Detection

Graph data science enables you to explore and analyze network structures using searches, queries and graph algorithms. Although graph data science draws upon graph theory, a subfield of discrete mathematics, you don't need an academic background to benefit from it.

Graph data science improves the accuracy of fraud predictions. Because fraud is costly and the scale of the problem so large, financial services firms using graph data science report that even a fractional percentage increase in accuracy drives millions of dollars in savings. A large Neo4j customer in the financial services industry reported finding tens of millions of dollars in fraud in just the first few months of using Neo4j for graph data science.

Improved prediction accuracy derives from graph analytics and graph feature engineering. Once data is connected in a graph database, it is possible to engineer graph features derived from connection-related metrics such as the number of relationships going into or out of nodes or a count of potential triangles or neighbors in common. Community detection algorithms highlight groups in your data so you can investigate possible fraud rings and dig deeper into unusual patterns.

Augmenting your current models with graph data science unlocks network structures in your data.

With graph data science, you detect more fraud in the *data you already have* without changing your ML pipeline. By analyzing historical data in this way, you may uncover fraud that is still recoverable, adding top-line benefits. Once you find patterns indicative of fraud, incorporate them in real-time operational fraud detection systems to stem such losses in the future.

Graph data science is effective when	Fraud example
Your data has many relationships	Credit score, co-signer, employer, identification number, credit line/credit limit
Your requirements change frequently	Evolving fraud tactics
Your data requires context	Multiple people with similar or the same characteristics, such as address, could be a group of loyal customers, a small fraud ring or a sign of synthetic identity fraud
Your data has patterns that are hard to see	Unusual behavior and suspicious patterns

Three Ways to Use Graph Data Science

To analyze the network structures in your data, load a copy of it into a graph database like Neo4j. You need data in a graph structure before you learn from the topology of your data and its inherent connections.

Here are three ways to use graph data science to find more fraud.

Graph Search & Queries for Exploration of Relationships

With connected data in a graph database, the first step is searching the graph and querying it to explore the relationships.

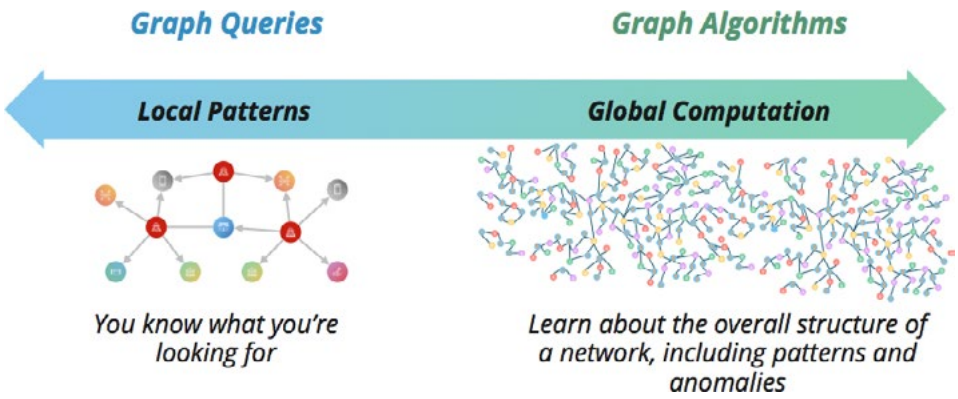
Domain experts might write a query to identify, for example, account holders with the same phone number, the same IP address or the same mailing address. A few lines in the Cypher query language replaces many lines of complex SQL code. Consider the sample Cypher query below that returns the shortest path from a start node to a close group of email or phone numbers. This query is a jumping-off point for several other analyses. If the starting node is fraudulent, nearby nodes may also be fraud or targets of fraud.

```
// 11 - Paths to Email and Phone Tokens
MATCH (p:P2P{transactionID:"startingID"})
OPTIONAL MATCH (n:P2P)-[:SENT_TO]->(phone:Phone)
WHERE phone.phoneNumber IN ['1stPhone','2ndPhone','3rdPhone','4thPhone']
OPTIONAL MATCH (n:P2P)-[:SENT_TO]->(email:Email)
WHERE email.emailAddress IN ['1stEmail','2ndEmail','3rdEmail','4thEmail'
WITH p, collect(n) AS nodes
UNWIND nodes AS n
MATCH path = shortestPath((p)-[*]-(n))
RETURN path
```

In addition, a data visualization tool like Neo4j Bloom enables you to visually explore graph datasets, to query data using code-free and pre-configured searches and to share visual perspectives with other teams.

Graph Analytics for Discovery

The second step is to use graph queries and algorithms to further investigate your data, get a sense of its structure and discover patterns and anomalies.



When you know what you're trying to find – such as identifying people are in a known fraudster's extended network – a graph query works well. However, when you know the general structure you're looking for but not the exact pattern, consider using a graph algorithm.

A graph algorithm is code specifically written to perform a certain type of sophisticated analysis, usually looking at a graph dataset as a whole.

After running a community detection algorithm, for example, a data scientist might hand off the results to an analyst to investigate whether the strong connections among account holders indicate a fraud ring, another fraud pattern or something else entirely, such as customer loyalty within a particular geography.

"As a math major, I never thought I would use the graph theory courses I took. Now, all of a sudden, it's most of what I do. My favorite part is the fact that the graph data structure incorporates all these mathematical concepts, and I can take advantage of them. I can do something that would require a lot of linear algebra in Spark or Python without any effort because the structure inherently builds it in."

– *Omar Azhar, Senior Manager Advanced Analytics, EY*

The Neo4j Graph Data Science Library includes enterprise scalable graph algorithms optimized to run against connected data in Neo4j. This library offers an enterprise-grade method for data scientists and analysts to run graph algorithms against connected data at scale.

Type of Graph Algorithm	Example Algorithms	Use in Fraud Detection
 **Community Detection**	Weakly Connected Components (Union Find), Louvain Modularity, Label Propagation	Identify disjointed groups that share identifiers. Identify communities that frequently interact.
 **Similarity**	Node Similarity using Jaccard	Measure account similarity or fraud ring similarity.
 **Centrality**	PageRank	Measure influence and transaction volumes.
 **Heuristic Link Prediction**	Common Neighbors	Find unobserved relationships and add them to your data.
 **Pathfinding & Search**	Shortest Path	Filter transactions with extremely short paths between people.

Graph Feature Engineering

Using graph algorithms and queries, data scientists find features that are most predictive of fraud to add to their machine learning models. For example, after using a community detection algorithm to find anomalies of tight communities that investigators have confirmed characterize fraud rings, we can then extract the relevant graph features of the full dataset into a classifier model to significantly increase predictive accuracy of existing fraud detection strategies.

The flexibility, scale and ease of use of the Neo4j's Graph Data Science Library allow data scientists to quickly experiment with multiple approaches to validate the most predictive features, before moving a model into production.

Graph features can be as simple as a node's community ID or a centrality metric, or may include more complex statistics describing the characteristics of a given community, proximity to known fraudulent accounts, similar relationship patterns to previously labeled fraud, or any other descriptive network attribute. Features can be saved as a node or relationship property, or can be streamed directly into another model development environment. As the graph is updated, algorithms can be rerun as needed while still leveraging the previously seeded results, to continuously retrain the graph-based model with maximum consistency and computational efficiency.

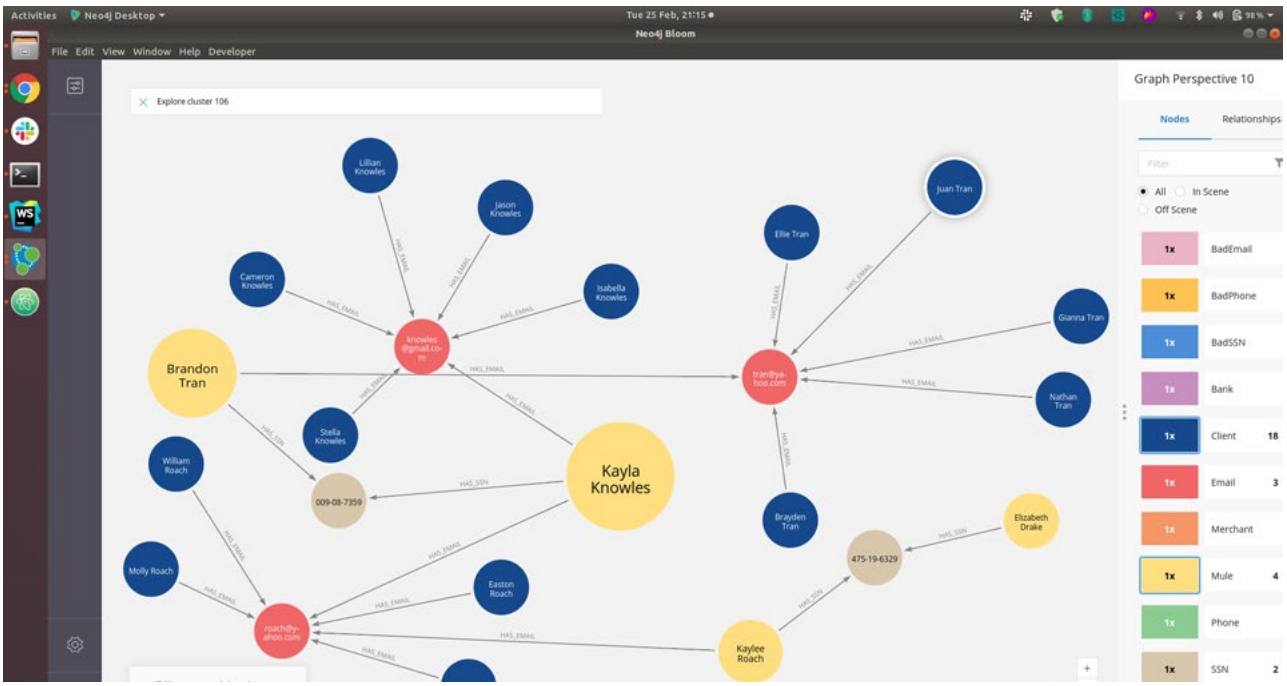
By leveraging graph features together with existing ML models and approaches, data scientists leverage the predictive power of network structure and relationships at scale without having to change validated and well-understood approaches.

We can demonstrate how topology can be predictive using a variation on the PaySim synthetic mobile transfer dataset. The following screenshot from Neo4j Bloom shows the results after running a community detection algorithm to find unusual islands of activity. We then use Betweenness Centrality to score the amount of influence each node has. In the visualization, the node size corresponds to the Betweenness Centrality score.

In this screenshot, we see a "suspicious" cluster (people sharing emails/phones/identifiers). A person with high betweenness centrality (large yellow) is more likely to be a mule.

"We drive performance in algorithms using feature engineering. A lot more education needs to happen about how to take an insight from the graph and turn it into features that then power algorithms or AI solutions. I really think the future is going to be using graph as a standard practice to accelerate innovation through machine learning and artificial intelligence."

- Neerav Vyas, Head of Analytics, Realogy Holdings



Visually exploring graphs helps analysts intuit what to investigate as well as what elements might be predictive. This type of structural information can be used for feature engineering and extracted to a table format for machine learning to predict whether someone is a fraudster. This is shown in the following table with a sample of clients, their betweenness centrality score, the number of identifiers they share with others, a weighted score based on what is being shared and then finally the prediction of whether they are a fraudster.

Client	Betweenness	Shared Identifiers	Weighted Shared Score	ML Model Prediction
William Roach	0	1	1	Normal
Kaylee Roach	32	2	4	Fraudster
Elizabeth Drake	0	1	20	Fraudster
Kayla Knowles	192	3	3	Fraudster
Nicholas Olsen	0	1	2	Normal

In the table, we see that William Roach shares an email address with several apparent family members, which is not very suspicious. Kaylee Roach is highly connected and shares multiple elements with others, so the model predicts she is a fraudster. Elizabeth Drake shares a social security number with Kaylee, and that red flag alone is enough to predict that she may be a fraudster.

Note that Nicholas Olson is not shown in the screenshot because of course a real network analysis would encompass far more clients than we see in this view.

Improving Fraud Detection with Graph Feature Engineering

Here are two examples of improving fraud detection using graph feature engineering: one for finding first-party and synthetic fraud and another for identifying fraud rings.

Example: Identifying First-Party Fraud

In first-party fraud, an individual (or group of people) misrepresents their identity or gives false information when applying for a financial product or service.

According to McKinsey, the fastest-growing type of first-party fraud is synthetic identity fraud. In synthetic identity fraud, the fraudster usually combines fake and real information to establish a credit record under a new, synthetic identity. This type of fraud results in major losses for financial institutions; an estimated 80% of all credit card fraud losses stem from synthetic identity fraud.

Organizations seeking to find fraud frequently have voluminous data to aid them in supporting investigations. However, with relevant data dispersed across relational database tables, data lakes and object storage, following the breadcrumbs across all of this data is arduous and time-consuming.

5 Steps for Finding First-Party Fraud using Graph Technology

The steps below are just one example. Your approach will vary depending on your goals and the data itself.

1. Create a graph of relationships of information about individuals. Connect all available information: account IDs, user names, account numbers, names, IP addresses, social media accounts, email addresses, identification numbers, mailing addresses, dates of birth and so on.
2. Consult with a fraud investigator to define what to look for. For example, consider:
   - Common attributes (same email address or phone number, for example)
   - Multiple parties using the same account
   - Short paths between transactions (a rapid return of a purchase with no support call or reason given, for example)
3. Run graph queries on these attributes or use similarity algorithms – like Common Neighbors – to investigate and then run community detection algorithms such as Weakly Connected Components (also called Union Find) to quickly look for disconnected islands of activity or Louvain Modularity to find groups that interact more with each other than the rest of the graph or network. Write the results back to your graph and notify investigators.
4. Investigators then use Neo4j Bloom to visually explore results and verify first-party fraud. Then they collect information to support a coordinated, rapid shutdown of anyone involved.
5. A more advanced approach is to convert graph algorithm scores into features to add to your machine learning model so that you identify more fraud faster and shut it down sooner.

Example: Identifying Fraud Rings

Fraud rings involve multiple parties working together to defraud merchants, banks or others.

Smaller rings are often run by a group of acquaintances or family members, some of whom may be unwitting participants. Large rings are more likely to be professional and more sophisticated, equipped with technology and resources unavailable to smaller rings.

Fraud rings may cross business roles, making them harder to detect since data about customers and vendors often resides in separate software systems or other data silos. A fraud ring could involve a buyer and a seller, many sellers and many buyers working together and even buyers and sellers with good reputations and valid transactions, with some fraudulent transactions mixed in.

Anomalies of many types may indicate fraud, from a sudden surge in sales and/or returns of a particular product, traffic from particular IP addresses or uncharacteristic purchases for a given demographic.

Although fraud rings are strongly linked, many businesses rely on manual or ad-hoc methods to detect them. Graph data science on connected data increases the likelihood of catching fraud rings in time to minimize – or eliminate – their impact.

BENEFITS OF USING GRAPH DATA SCIENCE

- Connections that identify millions of dollars in fraud
- Global view into connections between multi-source data
- Deep traversals that point to ever-shifting fraud behaviors based on complex connection patterns in the data

BENEFITS OF USING GRAPH DATA SCIENCE

- Graph-based representation that yields a robust feature set for finding fraud rings
- Faster identification of fraud ring patterns as well as similar patterns or variations on them
- Higher quality predictions by adding the behavioral dimension to your machine learning models

"Neo4j supported us in looking at very complex networks. Before we couldn't see very complex structures and connections at the fourth or fifth level. Neo4j enables us to see beyond the immediate customer or the immediate franchise we need to verify to their larger network. For us it was a very rewarding experience to actually see that there is something there. We kind of knew there was. We couldn't really prove it, but then we were actually able to see it."

- Julian Schibberges, Managing Director of the Bernstein Group

8 Steps for Finding Fraud Rings Using Graph Technology

The steps below are just one example. Your approach will vary depending on your goals and the data itself.

1. Use graph queries to uncover a suspicious pattern, such as multiple users coming from the same IP address. (Some of the techniques used in the first-party fraud example will also apply.)
2. Use community detection algorithms to identify strongly connected communities engaged in known fraud across various accounts using email addresses, phone numbers, authorized users and previously flagged activity.
3. Use the Louvain Modularity graph algorithm to examine whether hierarchies exist among these communities. Set thresholds to separate petty thieves from fraud rings so that investigators prioritize their efforts.
4. Use a centrality algorithm like PageRank to uncover influential individuals and to identify high frequency paths.
5. After verifying the pattern of one fraud ring, use a similarity algorithm such as Jaccard to identify other potential fraud participants and rings across your data.
6. Once the approaches to find fraud rings have been validated by investigators, and a labeled and scored dataset has been created, you can use these graph-based features in a machine learning pipeline.
7. Extract the calculated node and relationship properties – graph features from the previous step – into your ML environment (e.g., into a Python notebook). Join those properties with any other relevant tabular data. Use variable selection and model-building techniques to pinpoint the most important features and use them to predict future fraudulent activities or entities.
8. Once you're satisfied with your results, move your model into production. Write back any relevant findings to the Neo4j Graph Database to support further exploration.

Beyond Data Scientists: How Graph Analysis Benefits Fraud Investigators

Fraud investigators, like many criminal investigators, have a sense when something is just not right. Current fraud detection tools make it difficult for them to dig into an extended network since tabular data is not designed to capture relationships and reveal network structure.

Benefits of analyzing network structure extend beyond data scientists to everyone involved in fraud detection and investigation. Graph data visualizations are powerful for sharing results with the business and enabling further exploration of a connected dataset. After a data scientist runs graph algorithms, they can visualize the results in Neo4j Bloom. Analysts can then use codeless searches and easy-to-use interactions to explore the dataset in Bloom.

Conclusion

Fraud is a connected data problem.

Graph data science enables you to uncover more fraud and shut it down quickly. Accurate identification of fraudulent patterns focuses your time and energy on real fraud, significantly reducing effort on false positives.

Graph data science enables you to answer questions you cannot answer today without a tremendous amount of effort. The Neo4j Graph Data Science Library offers an enterprise-ready toolset for running sophisticated graph algorithms on connected data at scale. Graph analytics and feature engineering both add highly predictive relationships to your machine learning for better results.

Best of all, adding graph data science to your fraud detection toolkit is non-disruptive. Your existing ML models are already uncovering fraud. By analyzing your data as a graph, you add new dimensions and improve model accuracy without changing your existing ML pipelines. At the same time, you harness the power of graph algorithms to analyze the network structure of your data.

The more fraud you find, the more effective your teams will become at detecting even more subtle cases of fraud (in a virtuous cycle). You uncover a pattern, pursue it, follow another lead and find more anomalies. You operationalize those patterns to detect in real-time and explore yet again to find the ever-changing tactics that fraudsters use to evade detection. At the same time, experience with graph analytics in one area leads to deployment in other areas, such as managing risk and compliance, master data management and real-time recommendations.

Neo4j provides the first enterprise-grade approach to data science that harnesses the natural power of relationships and structures to infer behavior. It offers an integrated graph database for graph persistence so there's no need to recreate your graph each time it changes; your graph data is natively stored in the graph, available for further exploration and visualization. Neo4j enables you to scale to tens of billions of nodes, empowering you to move from a valuable initial proof-of-concept to a high-performance production environment.

Using Neo4j for graph data science, you gain a practical approach to increase your predictive accuracy with the data you already have.

To learn more:

- Get a sample chapter from the O'Reilly Graph Algorithms ebook
- Try out the Graph Data Science Sandbox

Questions about Neo4j?

Contact us around the globe:
info@neo4j.com
neo4j.com/contact-us