Suspicious events findings using programmatical implementation of the algorithm

During the process of the project, I have been working on implementing knowledge graph algorithms in Java source code. By accomplishing this task, we now have a programmatical way of connecting and applying knowledge graph algorithms to the graph database. The goal of this procedure is to add more flexibility and freedom when applying the algorithms on the data. Through NEuler (neo4j interface for graph algorithms), we had very limited power and flexibility. Now we can:

- Modify the algorithm's implementation
- Create our own unique recipes
- Combine multiple/various algorithms together
- Apply the algorithm on more specific data intervals

This document will contain insights and findings that were suspected as abnormal behaviours during the experimentation of the programmatical implementation of the knowledge graph algorithms.

From the first iteration of anomaly detection, it was suggested to experiment the post and pre fraud behaviour of company Marvell. Because we know that company Marvell has committed fraudulent activities in 2015, we can examine their behaviour before fraud and after fraud to explore fraudulent patterns. In addition to this, we obtain prior knowledge from machine learning algorithms. Various techniques were applied on the data and certain data points were classified as anomalies. We can now use the prior knowledge to examine our results with bias being included.

Both above mechanisms will be applied, and the resultant insights will be in the below document.

Part one of Experiments

We will apply the same experiments with the ones found in document one. However, we will apply them on pre/post fraud time intervals.

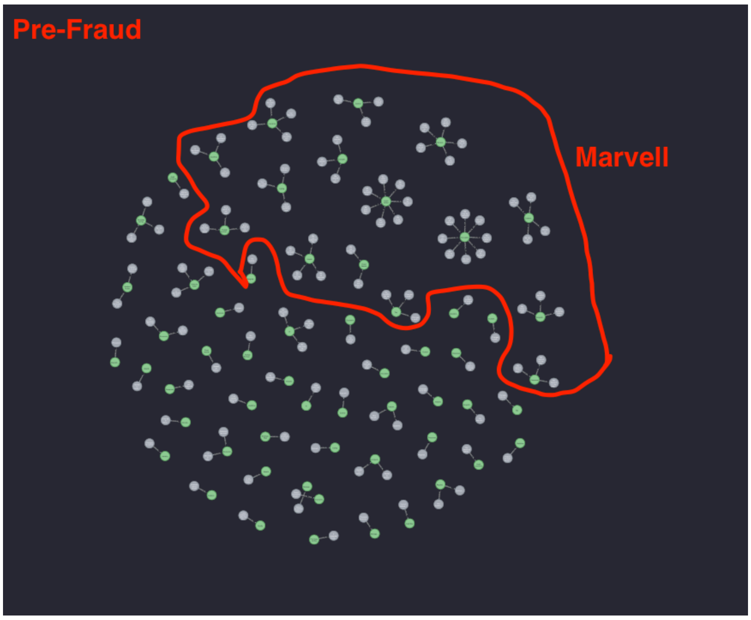
- Pre-Fraud = 2013, 2014, 2015
- Post-Fraud = 2016, 2017, 2018

[1] Annual Report & Risks

Experiment [1.1] Parameters applied with the Algorithm

- Algorithm: Weakly Connected Components and Degree Centrality
- Relationship: ANNUAL_CHARACTERISE

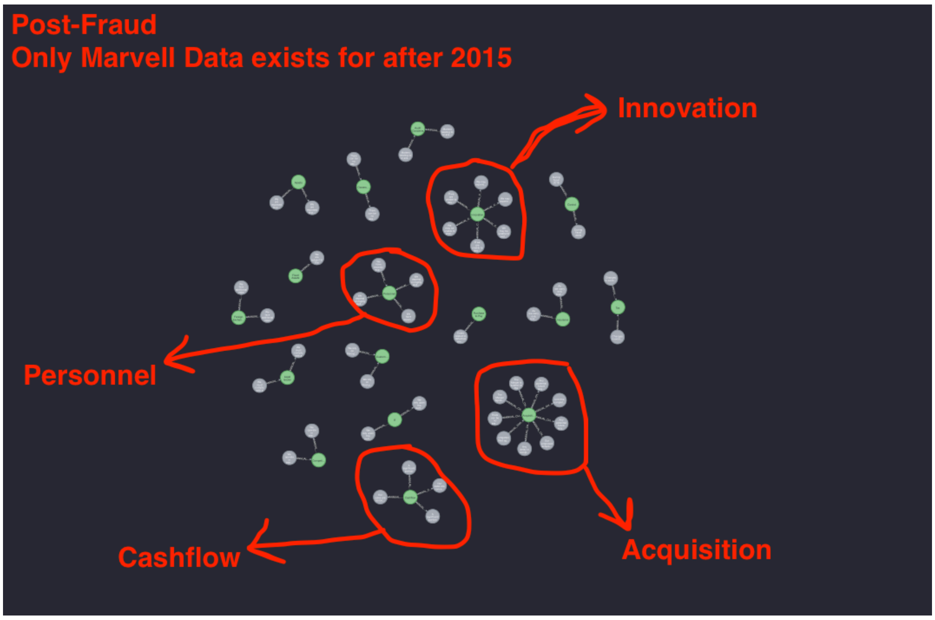
Pre-Fraud Results



Node	Degree Power
Areas – Innovation - **Marvell**	8
Areas – Legal Issues - **Marvell**	7
Areas – Ownership - **Marvell**	5
Areas – Insurance – **Marvell**	4
Areas – Customers – **Marvell**	4

- Algorithm: Weakly Connected Components and Degree Centrality
- Relationship: ANNUAL_CHARACTERISE

Post-Fraud Results



Node	Degree Power
Areas – Acquisition - **Marvell**	9
Areas – Innovation – **Marvell**	6
Areas – Personnel – **Marvell**	4
Areas – Cashflow – **Marvell**	4
Areas – Legal Issues- **Marvell**	2

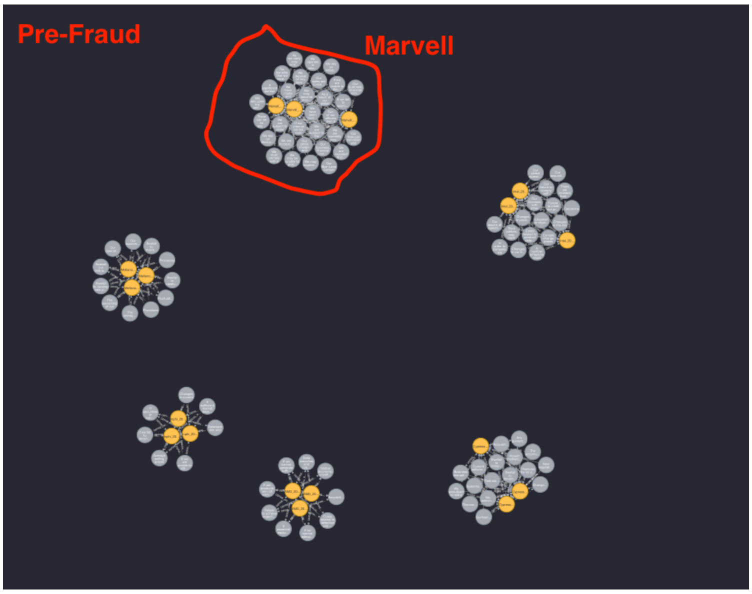
In this use case, we can observe that post-fraud period there is only data associated with ***Marvell***. This is an indication of data quality and extraction anomaly.
From the ***Degree Centrality*** algorithm, we can observe that the degree power of the most popular nodes has changed. This change could be a case of ***Marvell*** becoming more conscious of fraudulent activities, or it could be a case that the fraudulent activities were associated with the high degree nodes from pre-fraud (Area-Innovation and Area-Legal Issue). Area-Acquisition was not even in the top-5 popular nodes in pre-fraud period and now it is the most popular in post-fraud, while Area-Legal issue was the 2nd most popular in pre-fraud it now just has power of degree 2.

Experiment [1.2] Parameters applied with the Algorithm

Parameters applied with the Algorithm

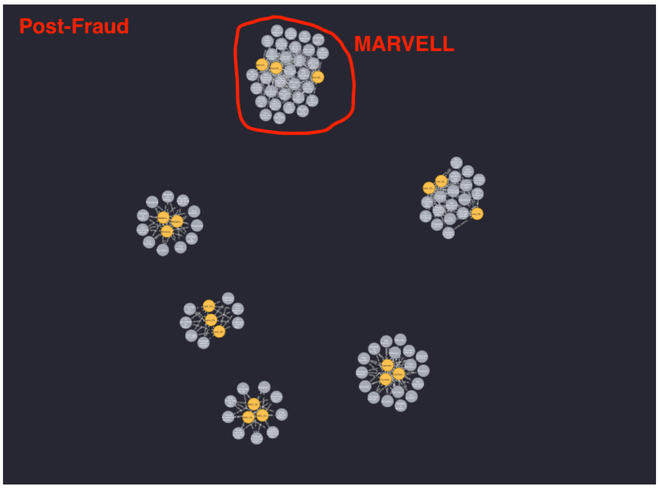
- Algorithm: Weakly Connected Components & Degree Centrality
- Relationship: DEFINE_ANNUAL_RISKS

Pre-Fraud Results



Node	Degree Power
AnnualReport – 2013 - **Marvell**	31
AnnualReport – 2015 - **Marvell**	31
AnnualReport – 2014 - **Marvell**	31

Post-Fraud Results



Node	Degree Centrality
AnnualReport - 2016 - **Marvell**	31
AnnualReport - 2017 - **Marvell**	31
AnnualReport – 2018 -**Marvell**	31

From this query we can observe that for every year in our dataset company ***Marvell*** are defining the same number of risks. This pattern has not changed however, from the previous example we can see that the risks and reports associated with different Areas has changed.

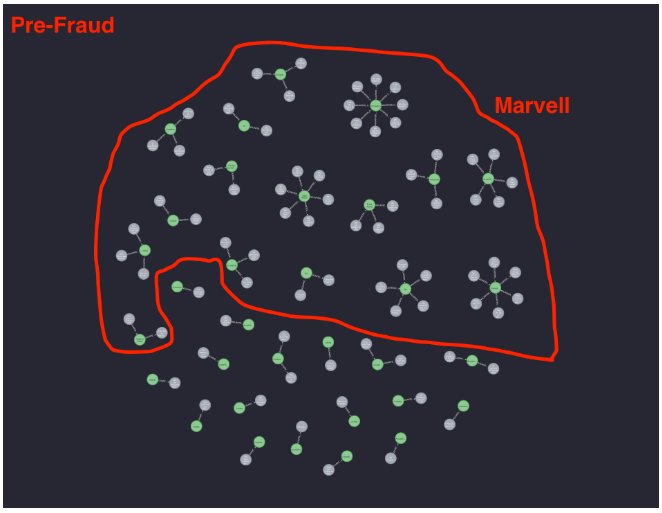
[2] Quarter Report & Risks

Experiment [2.1] Parameters applied with the Algorithm

Parameters applied with the Algorithm

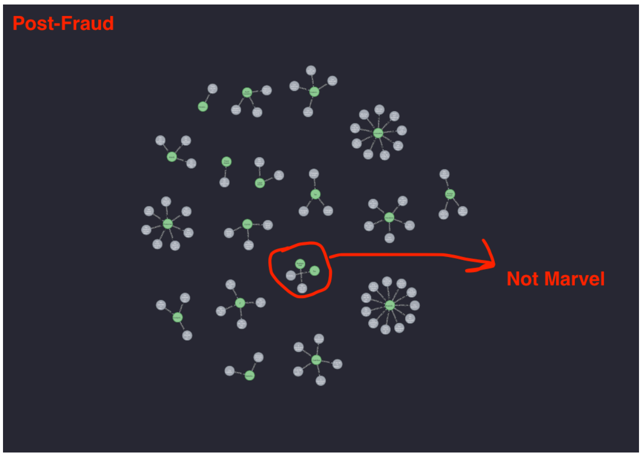
- Algorithm: Weakly Connected Components and Degree Centrality
- Relationship: QUARTER_CHARACTERISE

Pre-Fraud results



Node	Degree Power
Areas – Innovation – **Marvell**	8
Areas – Legal Issues – **Marvell**	6
Areas – Ownership - **Marvell**	6
Areas – Acquisition - **Marvell**	5
Areas- Tax - **Marvell**	5

Post-Fraud results



Node	Degree Power
Areas – Acquisition – **Marvell**	11
Areas - Innovation - **Marvell**	9
Areas - Personnel - **Marvell**	7
Areas - Insurance - **Marvell**	5
Areas - Cashflow - **Marvell**	5

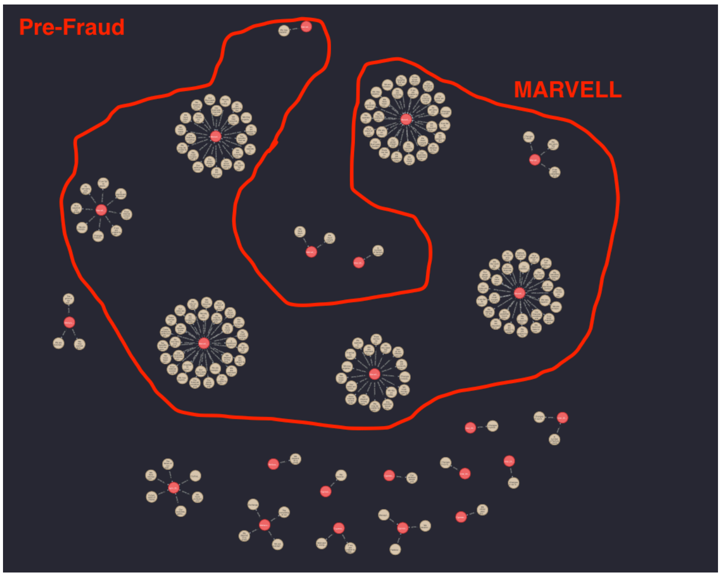
We expect a similar result as **ANNUAL_CHARACTERISE** and we indeed get one. We can see that in pre-fraud period Area-Acquisition is in the top 5 popular nodes from the **Degree Centrality** algorithm. This is an indication that it started rising in popularity going in the post-fraud period. On the other hand, we see that the popularity of Area-Innovation stayed the same, this is not the case in the **ANNUAL_CHARACTERISE** relationship. In the **QUARTER_CHARACTERISE** it increased by 1 and in the **ANNUAL_CHARACTERISE** it decreased by 2. This could be another indication of data quality and extraction anomaly.

Experiment [2.2] Parameters applied with the Algorithm

Parameters applied with the Algorithm

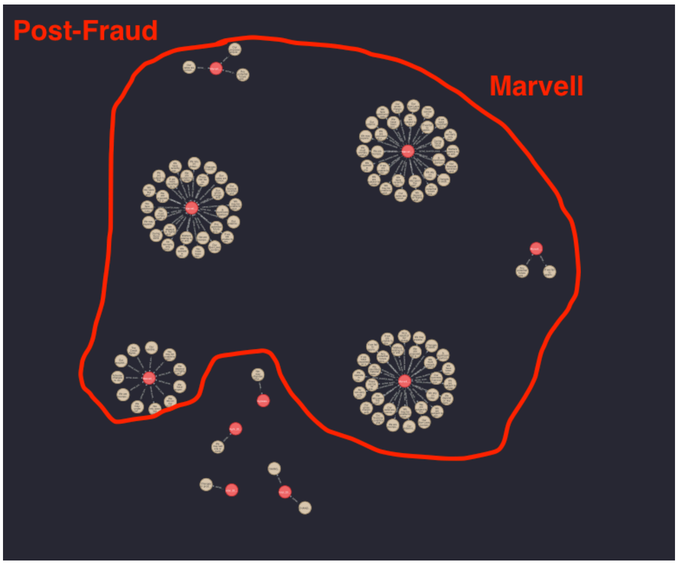
- Algorithm: Weakly Connected Components & Degree Centrality
- Relationship: DEFINE_QUARTER_RISKS

Pre-Fraud results



Node	Degree Power
QuarterReport – 02/2013 – **Marvell**	31
QuarterReport – 05/2013 – **Marvell**	29
QuarterReport – 05/2015 – **Marvell**	27
QuarterReport – 02/2014 – **Marvell**	22
QuarterReport – 02/2015 – **Marvell**	18

Post-Fraud results



Node	Degree Power
QuarterReport – 01/2016 – Marvell	30
QuarterReport – 01/2017 – Marvell	29
QuarterReport – 02/2018 – Marvell	28

Here we can observe that **DEFINE_QUARTER_RISKS** is not a constant across all quarter report, which was the case in **DEFINE_ANNUAL_RISKS.** From the degree centrality algorithm, it is also noticed that as year passes there is a very small decrease of number of risks being defined in the most popular quarter report of each year. They don't follow a same constant pattern as **ANNUAL RISKS,** but this could be because of the data quality and extraction issue that was noticed at **QUARTER_CHARACTERISE**. This could also mean that the risk being defined was not constant in the quarter reports in every year.

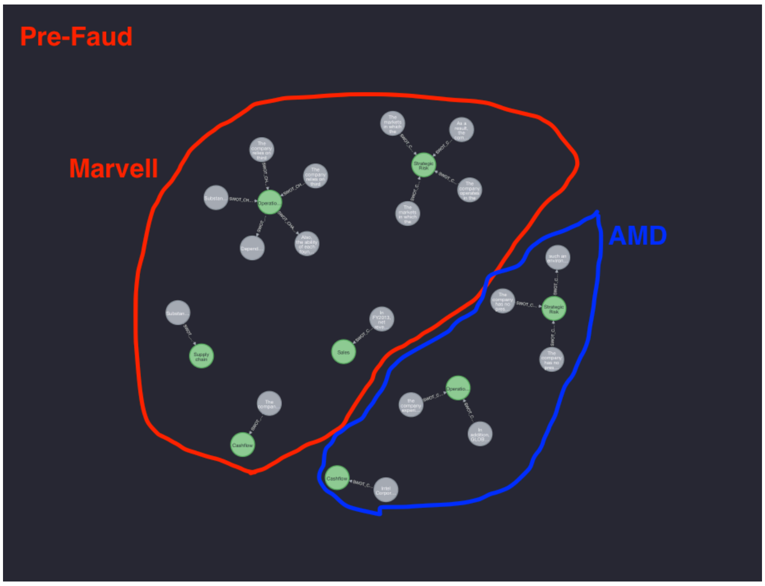
[3] Swot Report & Risks

Experiment [3.1] Parameters applied with the Algorithm

Parameters applied with the Algorithm

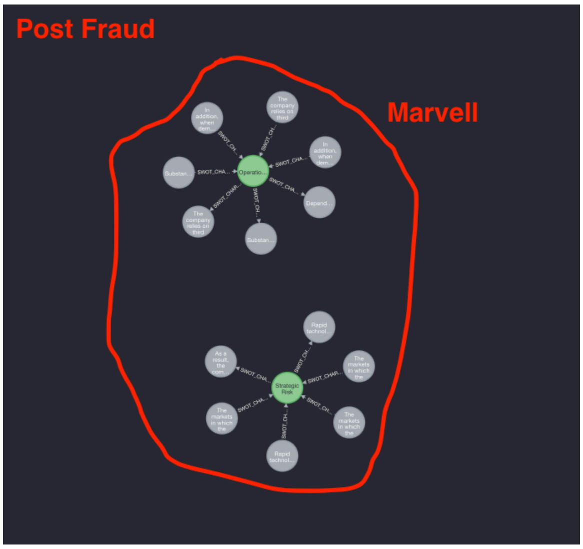
- Algorithm: Weakly Connected Components & Degree Centrality
- Relationship: SWOT_CHARACTERISE

Pre-Fraud results



Node	Degree Power
Areas – Operations – **Marvell**	5
Areas – Strategic Risk - **Marvell**	4
Areas – Strategic Risk – **AMD**	3

Post-Fraud results



Node	Degree Power
Areas – Operations – **Marvell**	7
Areas – Strategic Risk – **Marvell**	6

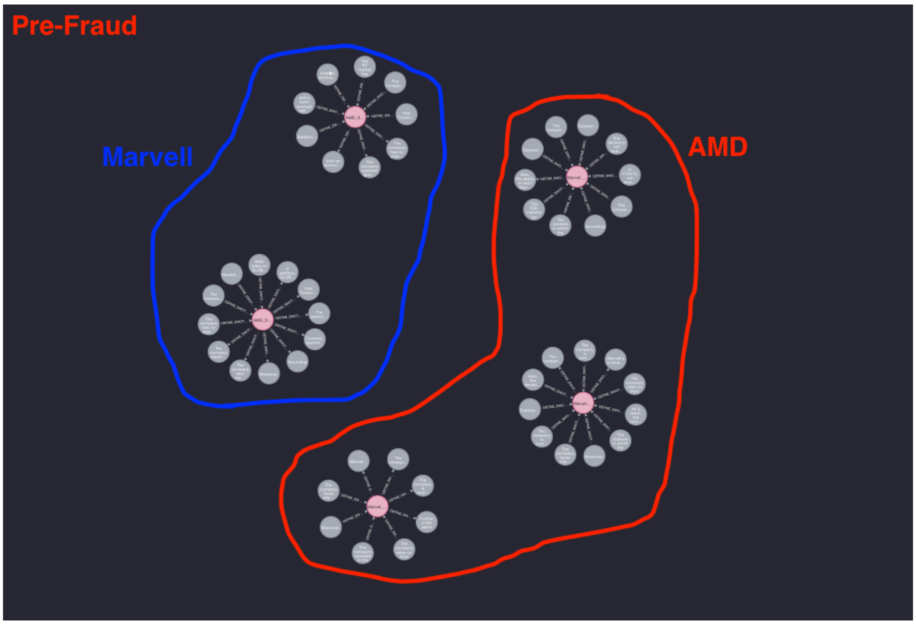
Here we can observe poor data quality, we do not have enough data to file a viable conclusion. The data in post-fraud time interval becomes even less.

Experiment [3.2] Parameters applied with the Algorithm

Parameters applied with the Algorithm

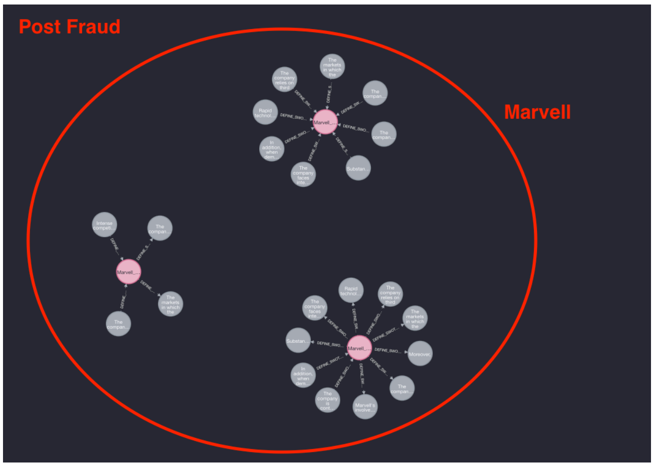
- Algorithm: Weakly Connected Components & Degree Centrality
- Relationship: DEFINE_SWOT_RISKS

Pre-Fraud results



Node	Degree Power
SwotReport – 2013 – **AMD**	13
SwotReport – 2015 – **Marvell**	11
SwotReport – 2013 – **Marvell**	10
SwotReport – 2014 - **AMD**	9
SwotReport – 2014 – **Marvell**	8

Post-Fraud results



Node	Degree Power
SwotReport - 2016 – **Marvell**	10
SwotReport - 2017 – **Marvell**	8
SwotReport – 2018 – **Marvell**	4

Here we observe a big decrease in risks being defines during post-fraud period. However, from Quarter and Annual reports this was not the case. This is again an indication of a data quality and extraction anomaly.

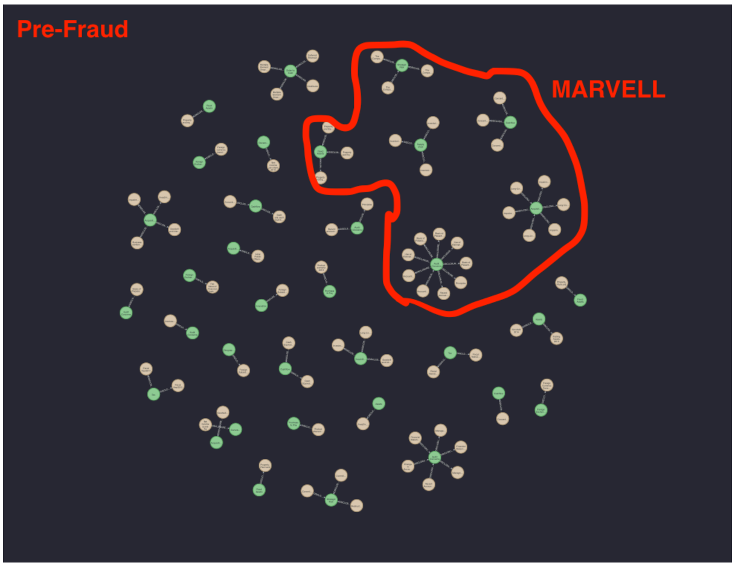
[4] Disclosures Involved

Experiment [4.1] Parameters applied with the Algorithm

Parameters applied with the Algorithm

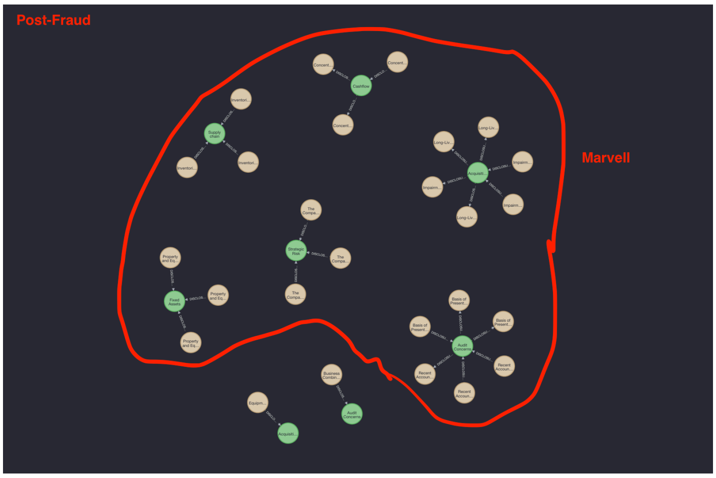
- Algorithm: Weakly Connected Components & Degree Centrality
- Relationship: DISCLOSURE_CHARACTERISE

Pre-Fraud results



Node	Degree Power
Areas – Audit Concerns – **Marvell**	9
Areas – Acquisitions – **Marvell**	6
Areas – Audit Concerns – **Cypress**	6
Areas – Order to Cash – **Inphi**	4
Areas – Supply Chain – **Marvell**	3

Post-Fraud results



Node	Degree Centrality
Areas – Audit Concerns – **Marvell**	6
Areas – Acquisitions – **Marvell**	6
Areas – Supply Chain – **Marvell**	3
Areas – Cashflow – **Marvell**	3
Areas – Strategic Risks – **Marvell**	3

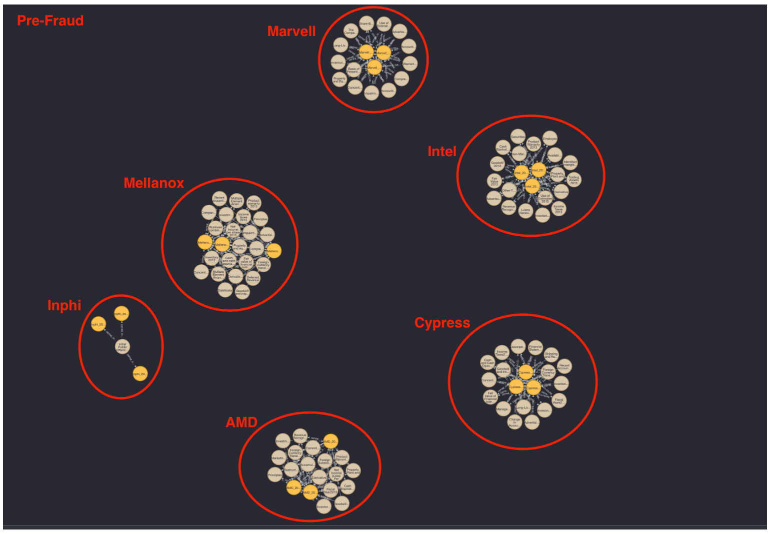
Here we can notice that ***Marvell*** pattern of characterising disclosure is the same in both periods post and pre fraud. The three most popular characterised Areas from disclosure are the same (Audit concerns, Acquisitions, Supply chain).
We can again observe here the data quality and extraction issue in post-fraud area. There not much data from the other companies, the extraction and quality were more focused on **Marvell**.

Experiment [4.2] Parameters applied with the Algorithm

Parameters applied with the Algorithm

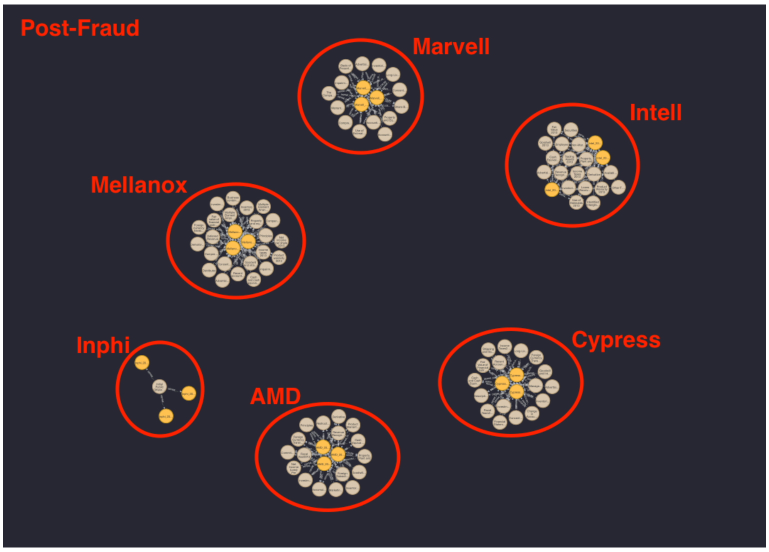
- Algorithm: Weakly Connected Components & Degree Centrality
- Relationship: DEFINE_DISCLOSURE

Pre-Fraud results



Node	Degree Power
AnnualReport – **Mellanox** (all 3)	23
AnnualReport – **Intell** (all 3)	19
AnnualReport – **AMD** (all 3)	17
AnnualReport – **Cypress** (all 3)	17
AnnualReport – **Marvell** (all 3)	14
AnnualReport – **Inphi** (all 3)	1

Post-Fraud results



Node	Degree Power
AnnualReport – **Mellanox** (all 3)	23
AnnualReport – **Intell** (all 3)	19
AnnualReport – **AMD** (all 3)	17
AnnualReport – **Cypress** (all 3)	17
AnnualReport – **Marvell** (all 3)	14
AnnualReport – **Inphi** (all 3)	1

With the last two experiments. (**DEFINE_DISCLOSURE, DISCLOSURE_CHARACTERISE**), we can notice something suspicious. Even though ***Marvell*** are defining less *Disclosure* than the other companies, they are characterising more *Areas* than them. This means that each of their *Disclosures* are describing multiple *Areas* simultaneously. This tactic makes the relationship of *Disclosure* and *Areas* more complex. This could be an indication that ***Marvell*** is making this relationship more complex on purpose with the goal to hide something.
The same thing can be observed with company ***Mellanox***, even though they are defining the largest number of Disclosures, they are characterising the least Areas. This means that the Disclosures are describing too few Areas. This tactic makes the relationship Disclosure and Areas too simple and there is not enough detail associated with them. This allows ***Mellanox*** to hide detailed information that could be fraudulent activity.

Conclusions from 1st exploration

- There exists bad data quality and extraction in post-fraud period.
- Complex relationship of Disclosure – Area from ***Marvell***.
- Too simple relationship of Disclosure – Area from ***Mellanox***.
- Same number of risks defined by ***Marvel*** but split differently in pre-fraud and post-fraud periods.

Part two of Experiments

Now I will apply modifications on the implementation on the algorithm and introduce the machine learning anomaly nodes in the procedure. We will use the machine learning anomalies to examine the structure of the communities. We will be able to examine the proportion of the community that is suspected as anomalous from the machine ***learning techniques.***

Modifications applied to the Algorithms

Label Propagation

The label propagation algorithm is a community detection algorithm. The pseudocode is as below:

1. Initialize the labels at all nodes in the network. For a given node x, Cx (0) = x.
2. Set t = 1.
3. Arrange the nodes in the network in a random order and set it to X.
4. For each x ∈ X chosen in that specific order, let Cx(t) = f(Cxi1(t), ...,Cxim(t),Cxi(m+1) (t − 1), ...,Cxik (t − 1)). Here returns the label occurring with the highest frequency among neighbours. Select a label at random if there are multiple highest frequency labels.
5. If every node has a label that the maximum number of their neighbours have, then stop the algorithm. Else, set t = t + 1 and go to (3).

This is the default implementation of the algorithm and the one used to find findings. We will now apply modifications on the implementation with the goal to improve the algorithm for our **specific USE-CASE**.

NOTE:

My java implementation returns the exact same results as Neuler only when the algorithm finds the result before the maxIteration stopping criteria. Label propagation has 3 different stopping criteria conditions implemented.

1. Stop IF no Label has been changed during iteration
2. Stop IF all nodes have same label as the maximum number of neighbours
3. Stop IF max iteration counter has been reached.

If the first two stopping criteria have not been satisfied and the result has yet not been found, the max iteration will be reached and the Neuler implementation compared to mine differ. This could be because the nodes are being propagated in a random order, this is impossible to duplicate in my java implementation.

We will introduce and include bias in our findings, we have prior knowledge of anomalies that have been detected from ***Machine Learning*** algorithms. We will modify the implementation considering the anomalies and the knowledge that ***Marvell*** is a company that has committed fraudulent activity in the past. The anomalous nodes are of node type *Ratio* or *Datapoint*, so we will need to apply queries and the knowledge graph algorithm on specific parts of the network. There are only three queries that can be directly used biased nodes in the structure.

1. Datapoint – Create -> Ratio

2. Datapoint – Characterises -> Area
3. Datapoint <- Contains – Annual Report.

These three queries will be our focus for this chapter of research, and we will focus on the specific years 2014 and 2015 where fraud was committed by Marvell. We will compare the structures with all the nodes and the structure with only the suspicious nodes.

[1] Change Random Shuffle

Instead of using a random shuffle on all the nodes and create a random ordering of propagation. We first will go through the nodes that are suspected as suspicious from the ***Machine Learning*** procedures. Once this mechanism is done, then we will go through the other nodes in a random order.

Result of this experiment:

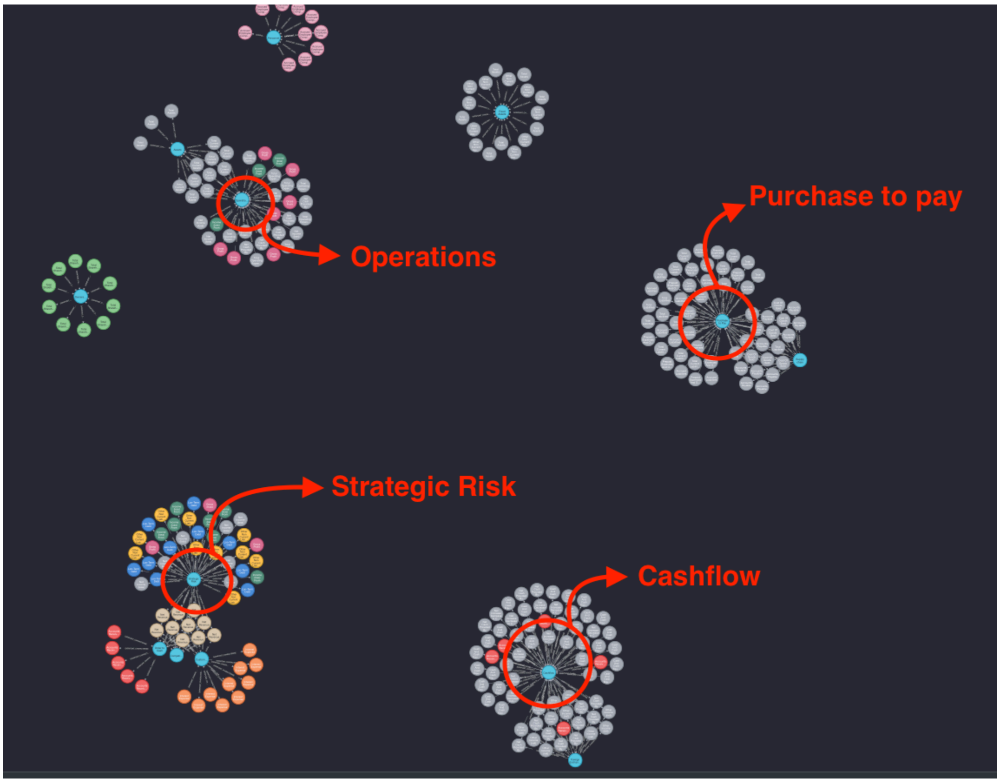
We get the same exact results as before; the reason is because the schema is manually human constructed and so we know how the communities will look like when all nodes are included. For this to work in a meaningful way we would need to accomplish one of the following.

1. Implement a similarity algorithm and use biased nodes as a similarity feature.
2. Modify graph-DB to separate structures when nodes are biased. Add more details to allow inspect the topology of the data and not only the structure.
3. Only use the biased nodes and do not include the rest.
4. Start the algorithm with all the anomalous nodes in the same community.
5. Explore results before stopping criteria has been met.

[2] Compare all nodes with biased nodes only

Query -- 1

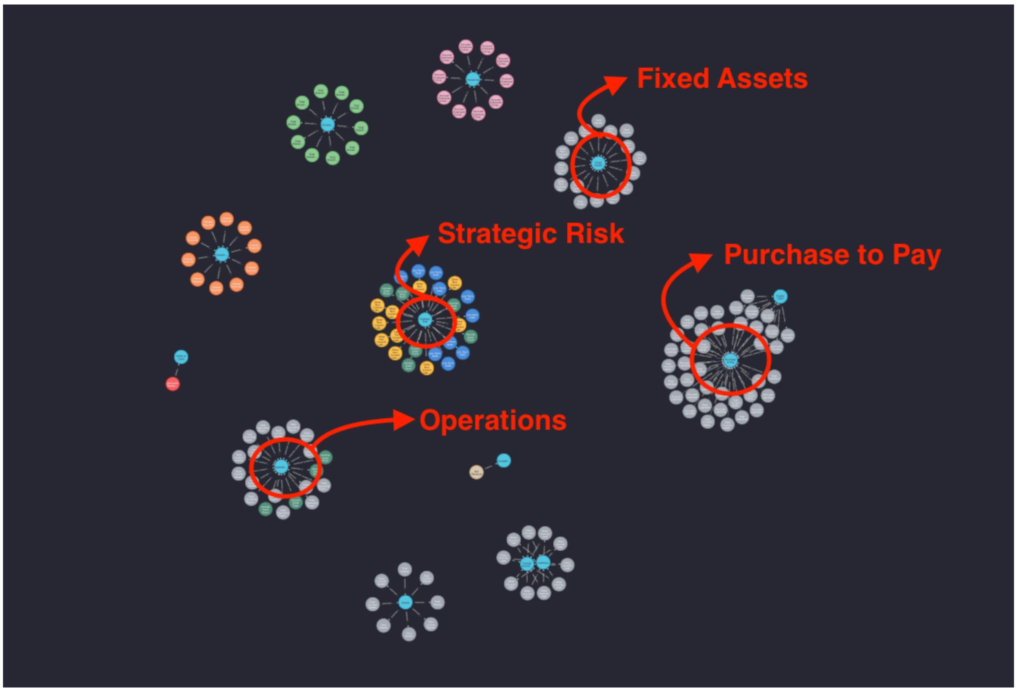
- Method: All nodes with introduced bias (normal + biased nodes)
- Relationship: DATAPOINT_CHARACTERISE



NODE	POWER

Area - Cashflow	69
Area – Purchase to Pay	60
Area – Strategic Risk	47
Area - Operations	39
Area – Foreign Exchange	20
Area – Supply Chain	20
Area - Customers	20
Area – Fixed Assets	17
Area – Order to Cash	15
Area - Assets	13
Area - Ownership	10
Area - Personnel	10
Area - Competition	10

- Method: Only biased nodes included
- Relationship: DATAPOINT_CHARACTERISE



NODE	POWER
Area – Purchase to Pay	40
Area – Strategic Risk	26
Area - Operations	20
Area – Fixed Assets	16
Area – Foreign Exchange	10
Area – Supply Chain	10
Area - Customers	10
Area - Personnel	10
Area - Ownership	10
Area - Cashflow	10
Area - Assets	8

Area – Order to Cash	1
Area - Competition	1

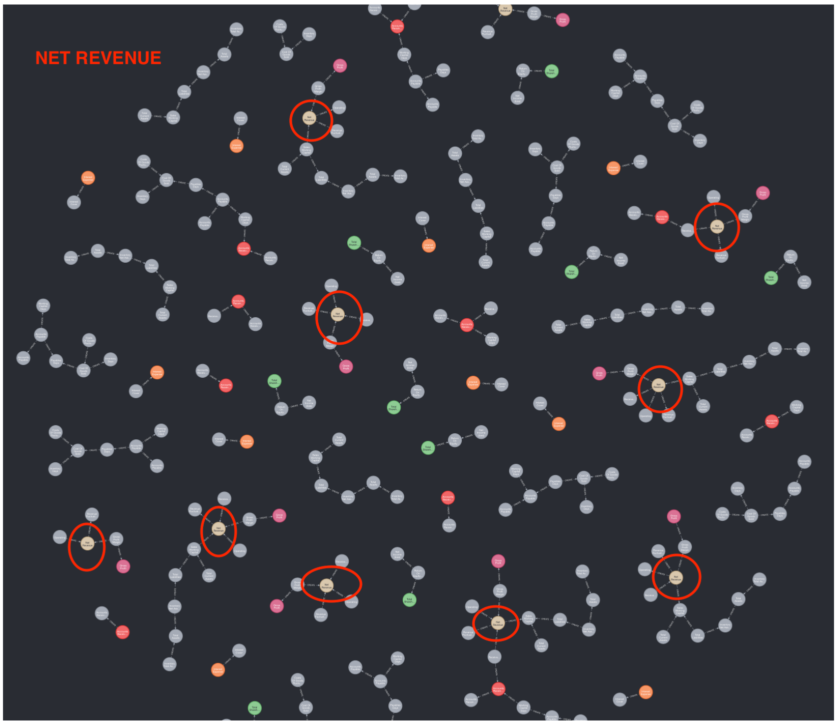
We will now compare the two tables and calculate the proportion of the biased nodes from all the data. This will help us see which communities are constructed mostly from abnormal detected nodes. This will be done by comparing the degree centrality algorithm result. The number indicates how many ***Datapoint*** nodes are related to the associated ***Area*** node. The first table considers all nodes relations, and the second table considers only the biased node relations. We can then compare and find the proportion of biased nodes when all nodes are included.

NODE	**POWER**
Area – Purchase to Pay	40/60 = **66%**
Area – Strategic Risk	26/47 = **55%**
Area - Operations	20/39 = **51%**
Area – Fixed Assets	16/17 = **94%**
Area – Foreign Exchange	10/20 = **50%**
Area – Supply Chain	10/20 = **50%**
Area - Customers	10/20 = **50%**
Area - Personnel	10/10 = **100%**
Area - Ownership	10/10 = **100%**
Area - Cashflow	10/69 = **14%**
Area - Assets	8/13 = **62%**
Area – Order to Cash	1/15 = **6%**
Area - Competition	1/10 = **10%**

From the most popular nodes from the knowledge graph found from ***Degree Centrality*** algorithm, I calculated the proportion of the popularity is constructed from ***Machine Learning*** anomalous nodes.
From the above table, we can observe the Areas of *Personnel*, *Ownership*, *Fixed Assets*, *Purchase to Pay*, *Assets* are related mostly with anomalous nodes. This could be worth to check if what was detected from the ***Machine Learning*** indeed something suspicious to consider.
Areas *Personnel* and *Ownership* communities are constructed 100% with anomalous nodes. Then Area *Fixed Assets* community is constructed 94% by anomalous nodes and Areas *Purchase to Pay* and *Assets* are constructed with more than 60% of anomalous nodes.

Query -- 2

- Method: All nodes including biased (normal + biased nodes)
- Relationship: CREATE



Node	Degree Power & Cluster Size
Datapoint - Net RevenueMarvell_2014_Q11	5 & 18
Datapoint - Net RevenueMarvell_2014_Q21	5 & 12
Datapoint - Net RevenueMarvell_2014_Q31	5 & 12
Datapoint - Net RevenueMarvell_20141	5 & 12
Datapoint - Net RevenueMarvell_2015_Q21	5 & 11
Datapoint - Accounts PayableMarvell_2015_Q31	3 & 9
Datapoint - Net RevenueMarvell_2014_Q41	4 & 8

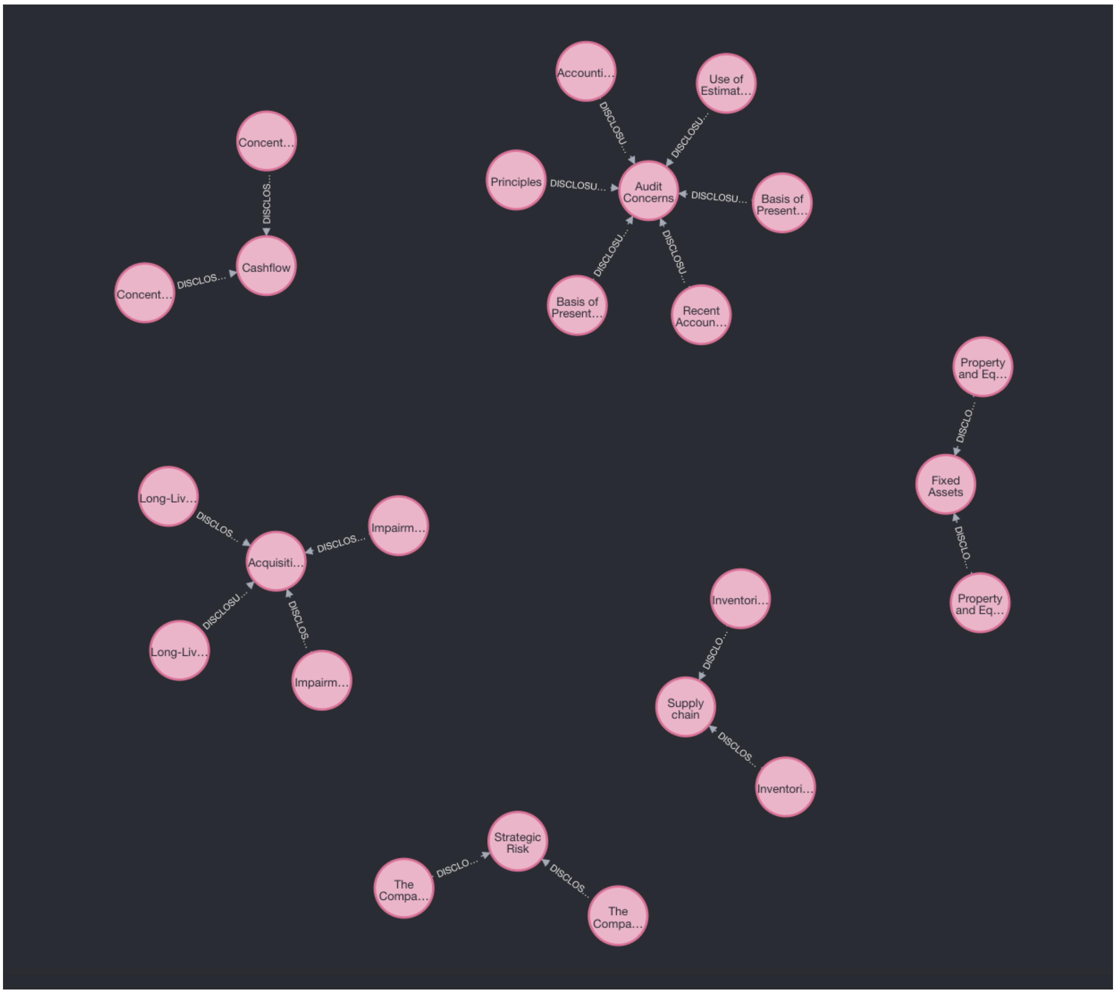
- Method: Only biased nodes
- Relationship: CREATE

For this query we get the same exact results as above, this indicates that all the communities that construct the knowledge graph are classified as anomalies from our ***Machine Learning*** algorithms. For years 2014 and 2015 for company **Marvell** all ratio and datapoint nodes are suspected as anomalies. In addition to this, we can observe that the larger sized clusters are from year 2014. In year 2015 it is noticed that even though the degree power of Area stays the same, the community sizes become smaller and this means less nodes are interconnected.

Query – 3

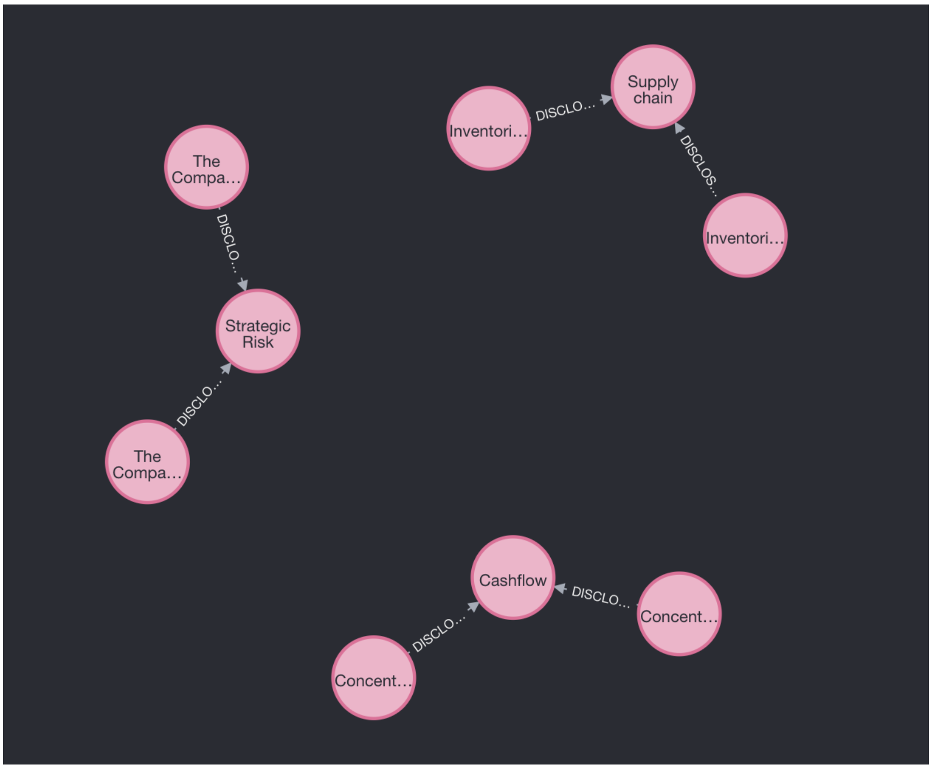
We will investigate Disclosures for years **2014 – 2015** in specific for **Marvell,** we will compare the raw results against the bias influence. The aforementioned means that only Areas that are related with bias nodes will be considered in the 2nd case. This will indicate which Disclosures are related with suspicious Datapoints. This task can be applied using simple cypher queries and there is no need to use a ***Knowledge Graph Algorithm***.

- Method: All data being considered (normal + biased nodes) in the specific time frame and company
- Inspecting: Disclosures related with Areas
- Query:
  Match (n:Disclosure) - [r:DISCLOSURE_CHARACTERISE] -> (m:Areas)
  WHERE n.CompanyUId ="1" AND n.Year = "2014" OR n.Year = "2015"
  Return n,r,m



The knowledge graph above shows that we have 18 Disclosures and 6 Areas for years 2014 – 2015 for company **Marvell.** We will now apply a query to inspect which of these Areas and Disclosures are related to ***Machine Learning anomalous*** Datapoints.

- Method: All data being considered in the specific time frame and company
- Inspecting: Disclosures related with Areas that are related with *Machine Learning anomalous Datapoints*
- Query:
  Match (n:Disclosure) - [r:DISCLOSURE_CHARACTERISE] -> (m) <- [rr:DATAPOINT_CHARACTERISE] - (p) <- [rrr:FOUND_ML_OUTLIER] - (i)
  WHERE n.CompanyUId ="1" AND (n.Year = "2014") OR (n.Year = "2015")
  Return n, r, m
  UNION
  Match (n:Disclosure) - [r:DISCLOSURE_CHARACTERISE] -> (m) <- [rr:DATAPOINT_CHARACTERISE] - (p) <- [rrr:FOUND_MS_AD_OUTLIER] - (i)
  WHERE n.CompanyUId ="1" AND (n.Year = "2014") OR (n.Year = "2015")
  Return n, r, m



The knowledge graph above shows that we have 6 Disclosures and 3 Areas for years 2014 – 2015 for company **Marvell** that are interrelated with anomalous datapoints**.** These nodes are indication where suspicious event might be occurring and need further exploration.
Very suspicious Areas that the investigation has found are *Strategic Risk*, *Supply Chain*, *Cashflow* for years 2014 – 2015 for company ***Marvell***.

Conclusion of Experiment two

1. 6 Disclosures are interrelated with anomalous Datapoints
2. Communities of Datapoint – CREATE -> Ratio becomes smaller in the latter year but keeps the same number of degree power.

3. For communities of Datapoint – CHARACTERISE -> Area, it was found that Areas *Personnel*, *Ownership*, *Fixed Assets*, *Purchase to Pay*, *Assets* are highly proportioned by anomalous datapoints.