

IMI BigData**AI**Hub Case Competition **Team 31**

Chenhao Zhu, Jenna Fu, Kecheng Xiao, Yuqi Liu, Zhengqi Sun

Table of Contents

01

Case Context

Define the question and clarify the case goal

02

Name Screening

Detect 50 Bad actors in our customer base using public data sources

03

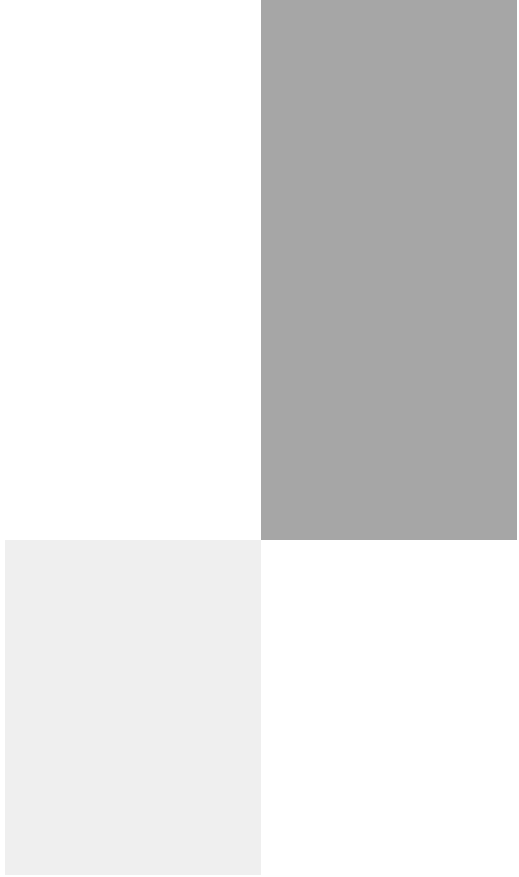
Risk Rating

Classify customers into Low, Medium and High risk and predict Bad actors

04

Client Connections

Add customer connections information to improve model



01

Case Context

How can we detect Financial Crimes?

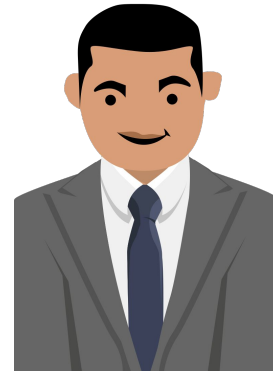
Suppose you want to reassess the likelihood for financial crimes for three existing clients of **Scotiabank**:



**Bezrukov, Sergery
Vitalyevich**



**Howard, Janice
Nelson**



Mark Stupar Lecy

Case Goal

- Identify whether or not these potential clients are high-risk individuals via **name screening**.
- Determine their initial risk rating at onboarding, by building a **risk rating model** with existing data.
- Monitor their transaction activities systematically to detect **interesting and abnormal networks**.



02

Name Screening

**Detect 50 Bad Actors in Current
Customer Base**

Task Overview

Customer Data

Name	DOB
Bezrukov, Sergey Vitalyevich	1973-10-18
Susan Gordon	1952-07-13
Romero, David	1945-09-30

Bad Actor Data

Name	DOB
Sergery Vitalyevich BEZRUKO	Oct-18-1973
Samantha Lewthwaite	1983-12-05
Marina Nikolaevna ZHEYNOVA	15/02/1985

Bad Actors in Current Database

Name	DOB
Bezrukov, Sergey Vitalyevich	1973-10-18
...	...

Underlying Assumptions:

1. We treat Name-DOB combination as each individual's unique identifier.
2. In cases where multiple DOBs are provided, we adapt the first one in the list.

Challenge #1: Inconsistent Names

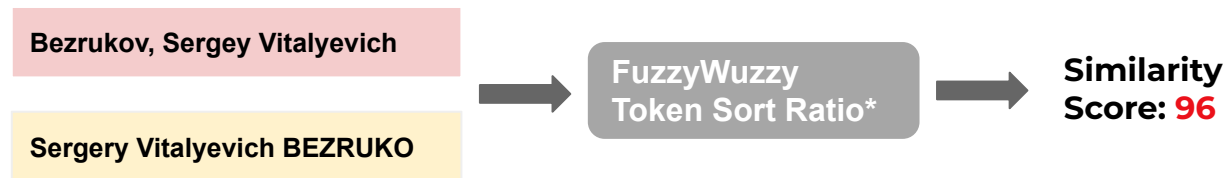
Problem Definition:

Customer and bad actor names may adapt different formats, therefore we cannot apply exact match to compare the two name lists.

Our Solution:

Apply **fuzzy matching** to compare the two name lists, with the *FuzzyWuzzy* package

Example:



* Token Sort Ratio: Sort the words in each string, and compares two strings using the **Levenshtein Distance**, producing a similarity score between 0 and 100.

Challenge #2: Large Datasets

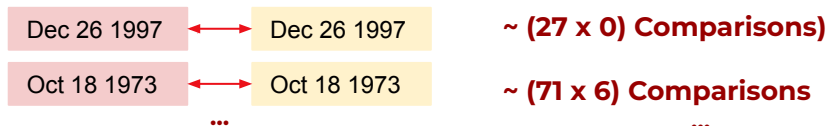
Problem Definition:

Customer Data and Bad Actor Data consist of thousands of entries, therefore difficult to compare every names (string) on the two lists.

~ (1M x 250K) Comparisons

Our Solutions:

1. For super faster matching, vectorizes the strings into **tf-idf** matrix to calculate **cosine similarity** between strings.
2. Partition both datasets by unique DOB, and compare data subsets with same DOB

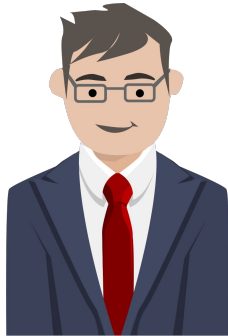


Task Outcomes

Cross-reference Bad Actor lists identified with:

- **Method 1: FuzzyWuzzy/Levenshtein Distance**
- **Method 2: String Grouper/TF-IDF + Cosine Similarity**


And generate final list of **50 Bad Actors**.



**Bezrukov, Sergery
Vitalyevich**

Recommendation

Since this assessed client falls within the watchlists, we suggest freezing their assets and accounts for further investigation.



03-A

Supervised Learning

Classify customers into Low, Medium and High Risk

The goals of building models

- **Assigning each client to a risk bucket Low, Medium or High**
- **Finding the most related features with the risk bucket**
- **Predicting the risk of each client being a bad actor or not**

Few steps before building model

Checking null values

Missing values can prevent the model from working

Changing data types

The types of data must be unified in order to support model

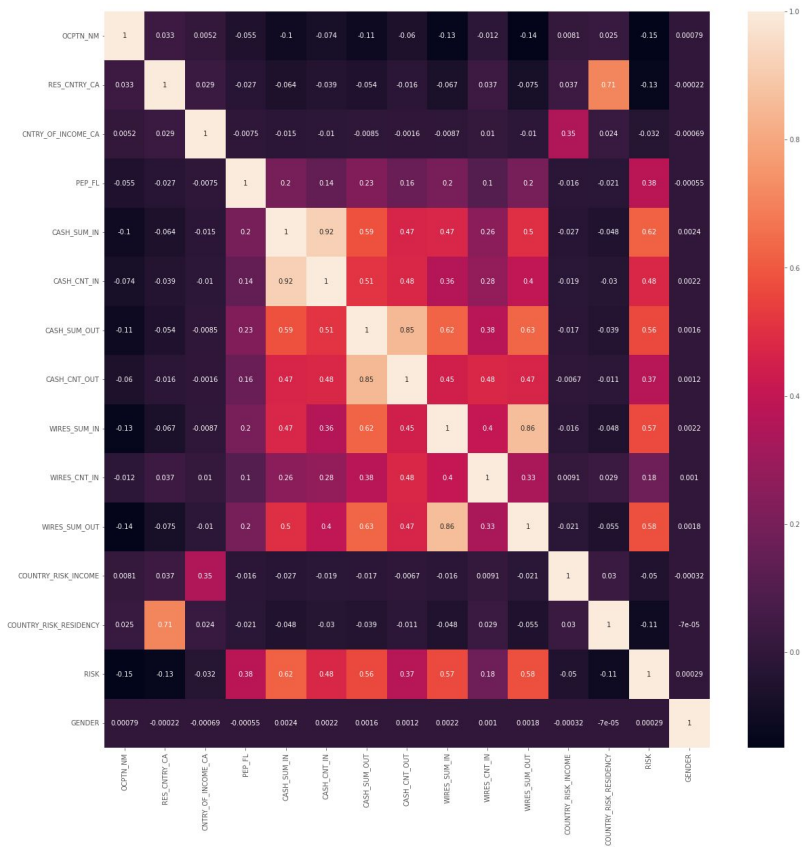
Decreasing the input size

Too many uncorrelated features will confuse the model and lead to low accuracy.

Train test split

Last step is separating the train dataset used to train the model and test the result.

Correlation matrix



The most correlated features with Risk are: ['PEP_FL', 'CASH_SUM_IN', 'CASH_CNT_IN', 'CASH_SUM_OUT', 'CASH_CNT_OUT', 'WIRES_SUM_IN', 'WIRES_CNT_IN', 'WIRES_SUM_OUT'] .

Hence those features will be selected to be the inputs of our following models.

70% of data are used for training model and 30% of data are used to test model.

Logistic Regression/RF/DT

Using 10-Fold Cross-Validation

K-fold cross-validation is a common technique used to evaluate the performance of a model. The basic idea is to split the data into K folds, and then train and test the model K times. In each iteration, the model is trained on K-1 folds and tested on the remaining fold.

In this situation, 10-fold cross-validation is used to provide a more accurate estimate of the model's performance. Here are the test results:

Logistic Regression/RF/DT

	Precision	Recall	Accuracy
Logistic Regression	0.9386366721457444	0.9501866666666666	0.9501866666666666
Random Forest	0.9450126495046363	0.95371	0.95371
Decision Tree	0.9375031332257667	0.9365266666666666	0.9365266666666666



03-B

Bad Actor Likelihood

Predict Bad Actors using results from Task 1 as
target variable.

Build Bad Actor Scoring

Using Estimation of Log-likelihood Function

$$f(X) = \frac{1}{1 + e^{-\beta X}}$$

- Has a natural range of 0 to 1
- With threshold, it is easy to adjust

Challenge: Imbalanced Dataset

Problem Definition:

This is an imbalanced dataset, with only 50/1000000 customers are labelled as bad actors.

Our Solutions:

Resampling

1. Oversampling the Bad Actor data (SMOTE)
2. Under-sampling the Normal Actor data

Issue: Overfitt the Bad Actor data

Model-level

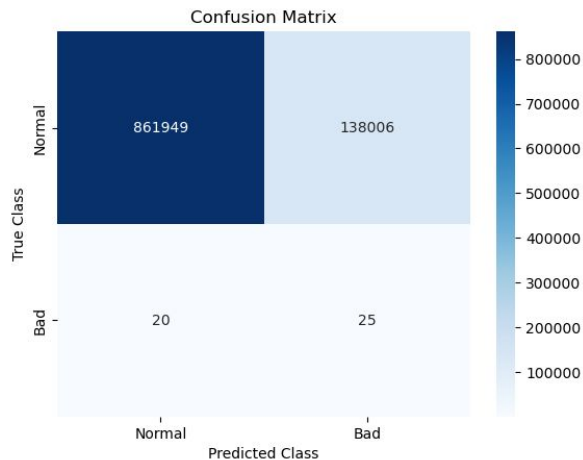


1. Use ML algorithm can naturally handle it (LR, DT, RF)
2. Adding weight to adjust the loss function

The approach in sklearn is:

```
LogisticRegression(class_weight='balanced')
```

Likelihood Result



Metrics

Accuracy	95.67%
Precision	99.99%
Recall	95.67%

*evaluate in 10 fold average, the precision and recall are weighted

04

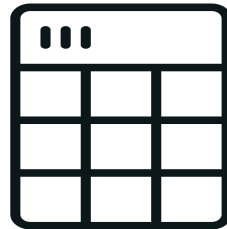
Client Connections



Task Approaches

Engineer New Features

Incorporate new features generated with graph data into the original feature sets.

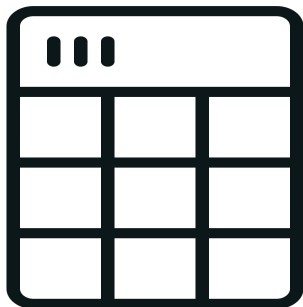


Customer Network Visualizations

Visualize and detect interesting network, finding criminal groups and cash flow



Engineer New Features



Original Features

+

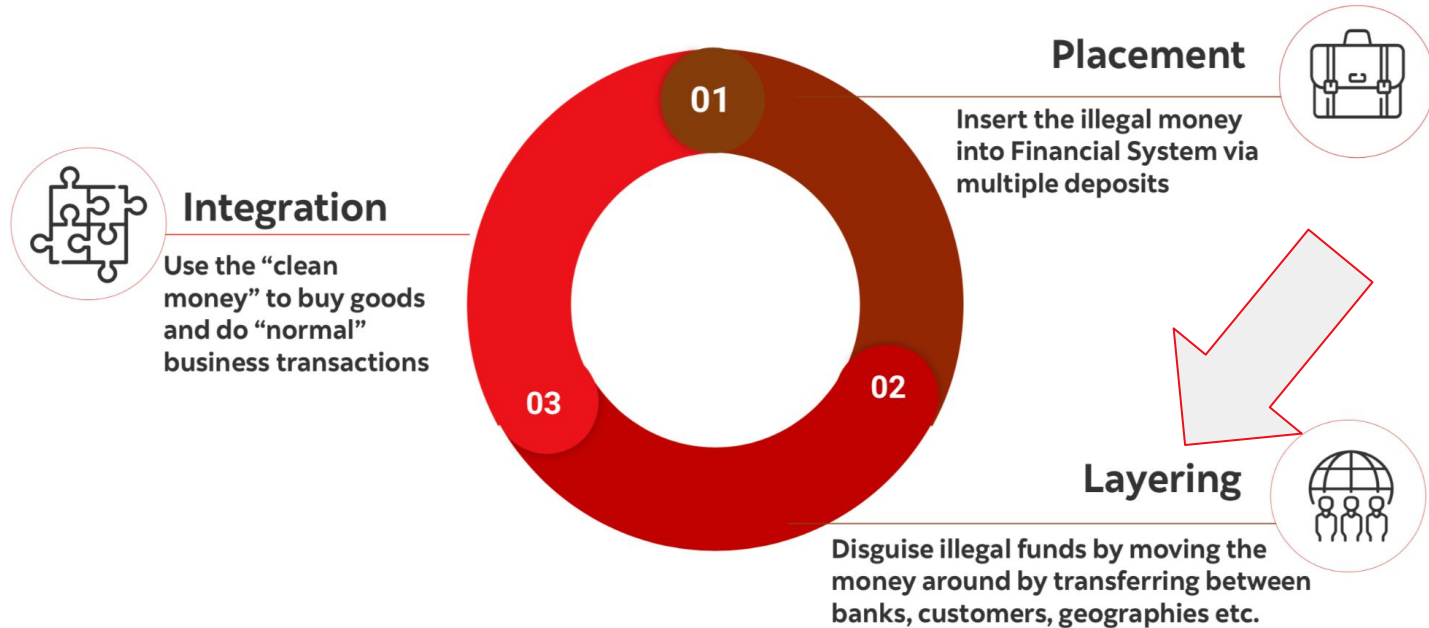
EMT_SUM_IN	Total Amount of EMTs Sent
EM_SUM_OUT	Total Amount of EMTs Received
OUTGOING_BA	# of Transaction TO BA
INCOMING_BA	# of Transaction FROM BA

New Features

Main Outcomes

1. Best model for Task 2A (RandomForest) increase in accuracy by ~0.1.
2. No significant improvement in best model for Task 2B.

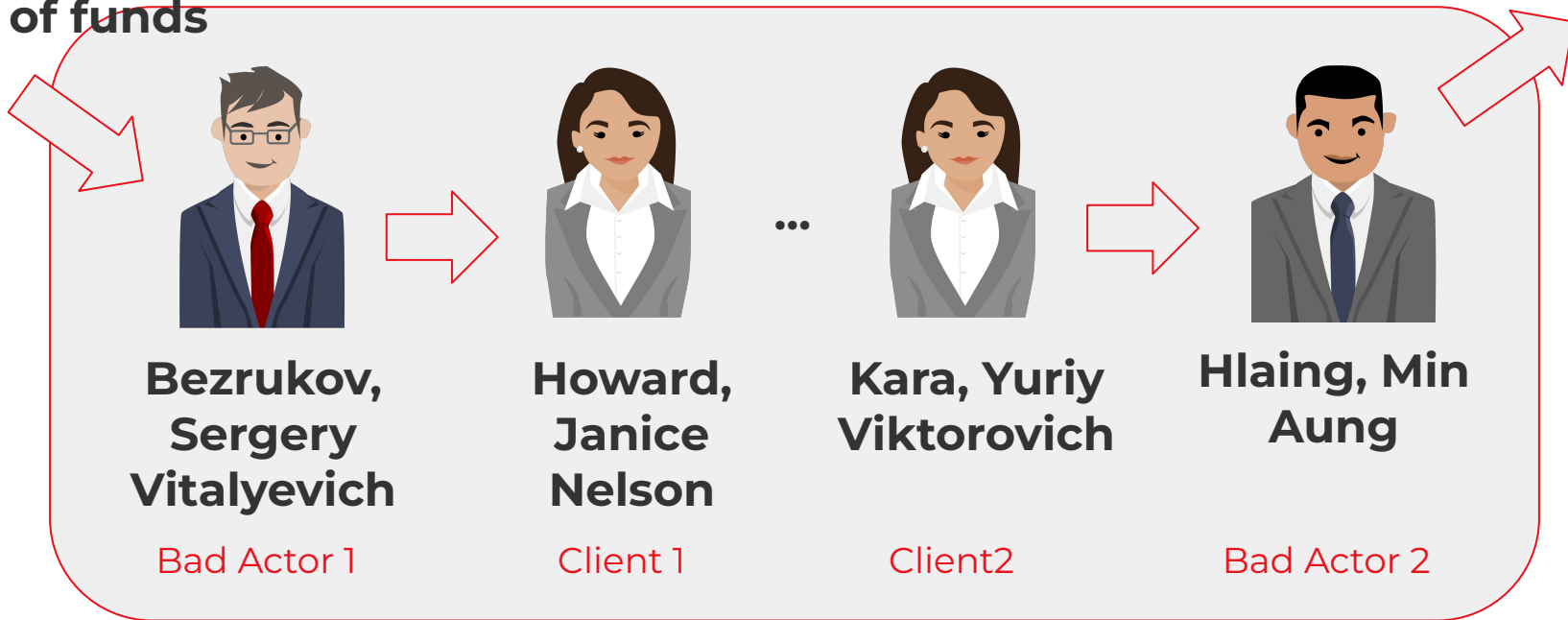
Why would we need to do client connections?



Why would we need to do client connections?

Unknown source
of funds

Legal money



The goals of Visualization

- **Finding if there is n-layers connections between two bad actors and figure out big anti-money laundering groups**
- **Present the cash flow and track the starter and destination**
- **Finding the core clients who have transaction with founded big criminal groups and strengthen supervision**

Customer Network Visualization



Risk level

Predicted risk level in
“low”, “medium”, “high”

+



Transaction Data

Cash Flow between
two clients with
frequency and amount

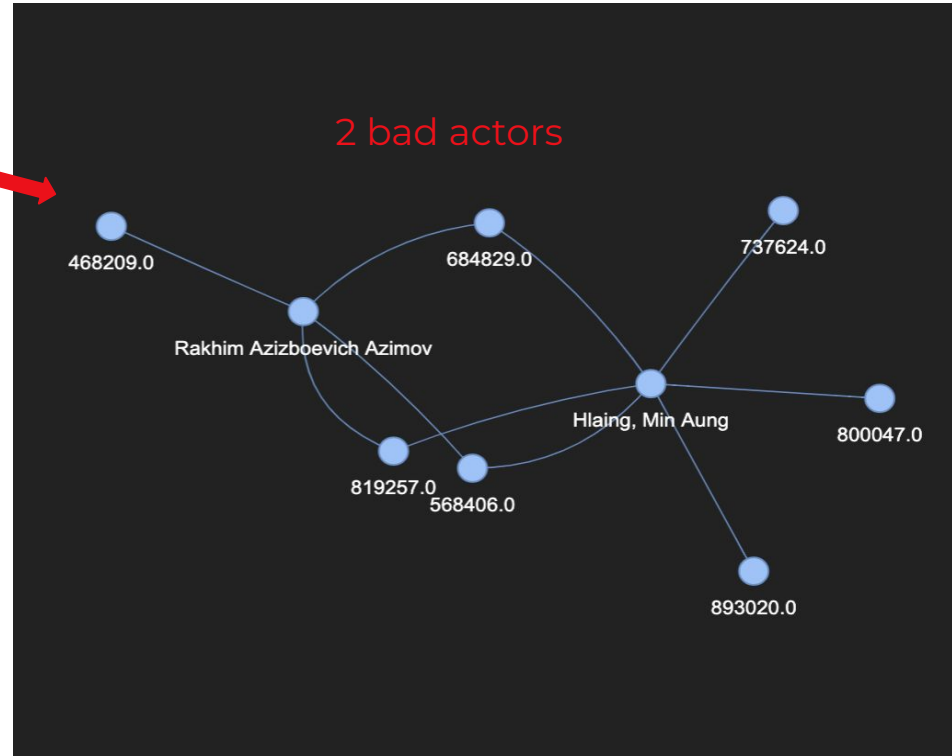
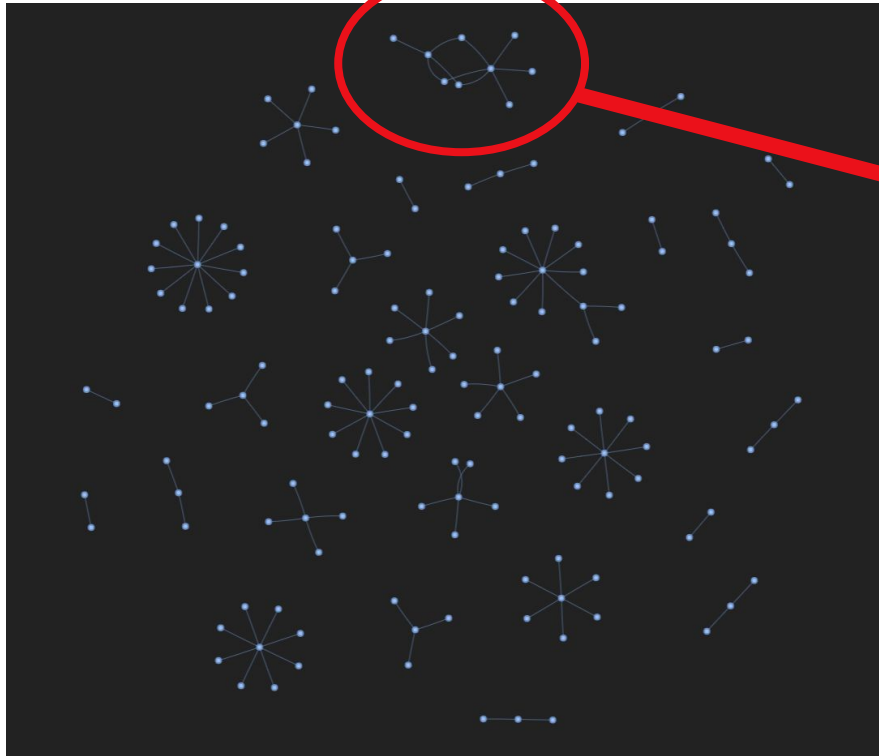
=



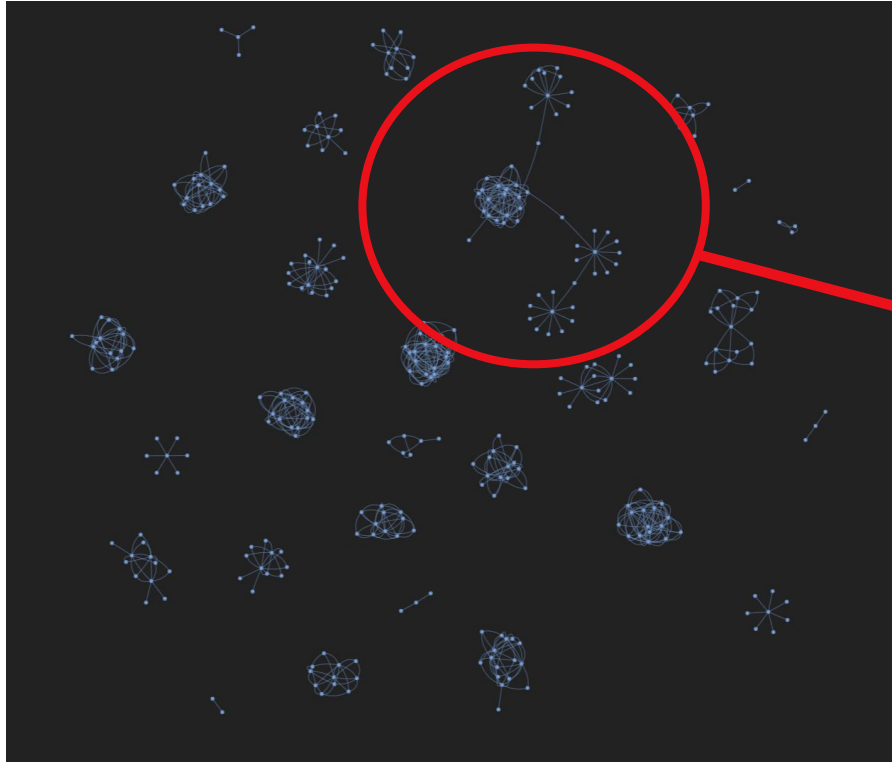
Network

Find big criminal
groups between
different bad actors

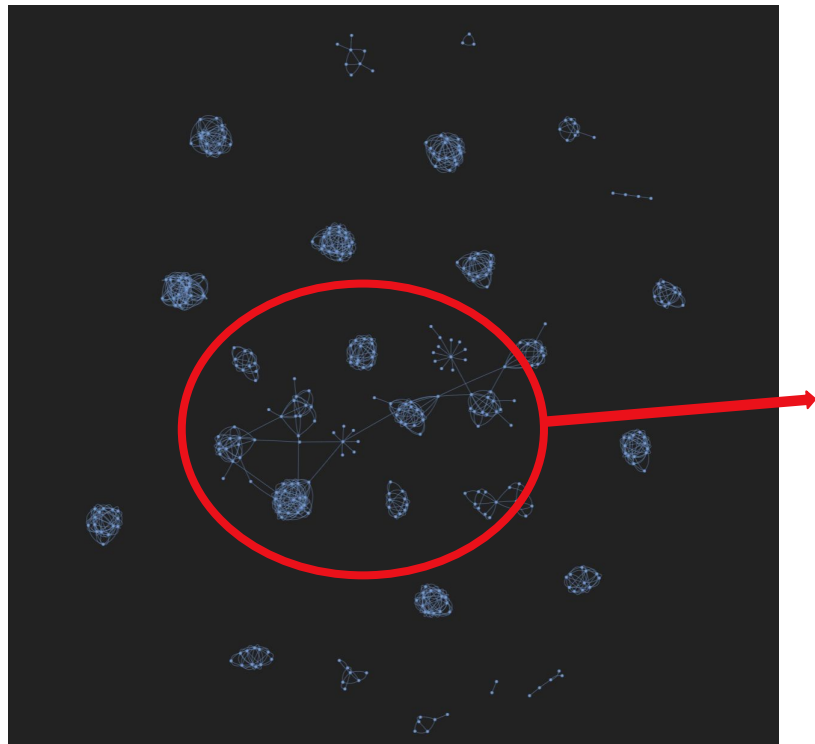
1-layer visualization model output



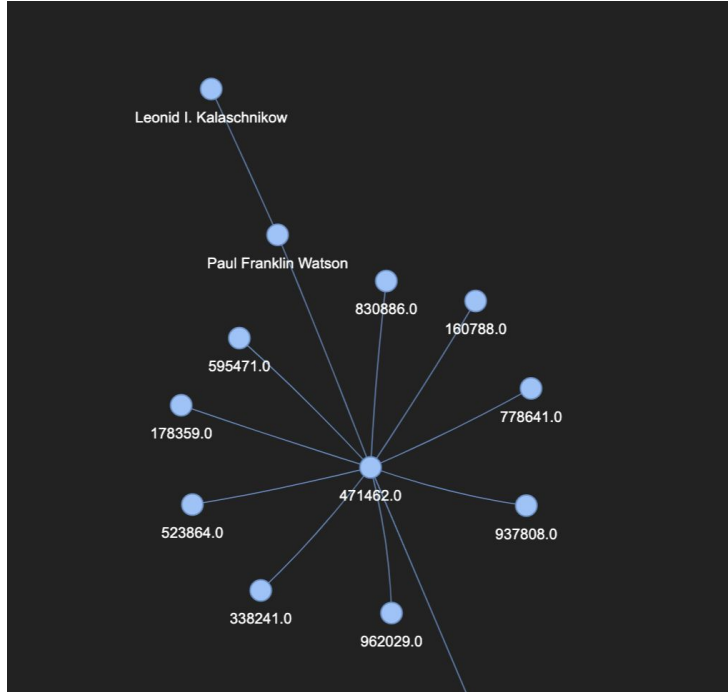
2-layer visualization model output



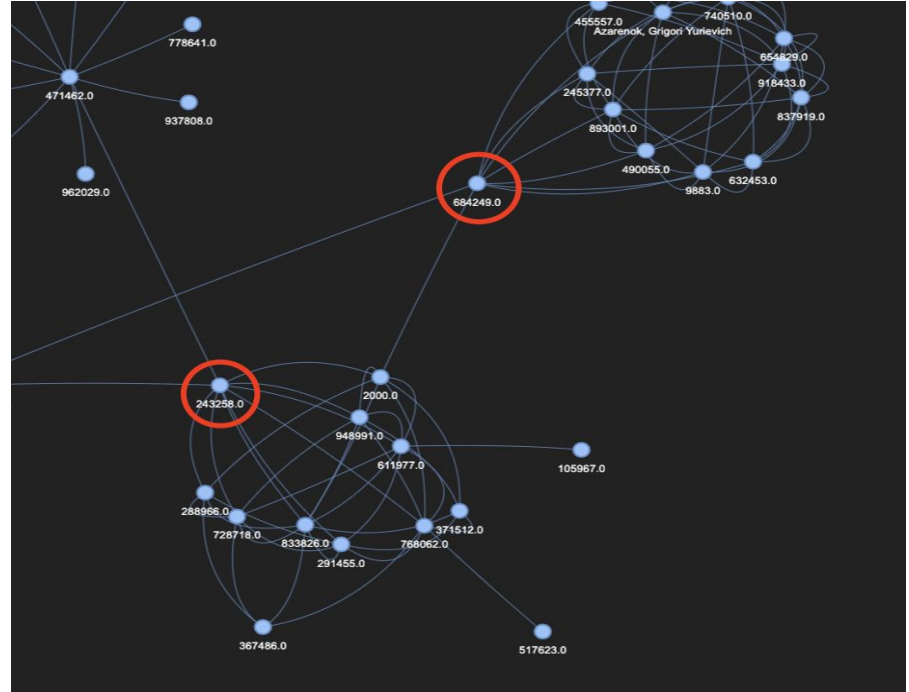
3-layer visualization model output



3-layer visualization model output



Starter or destination

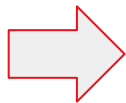


central clients who is not bad actor

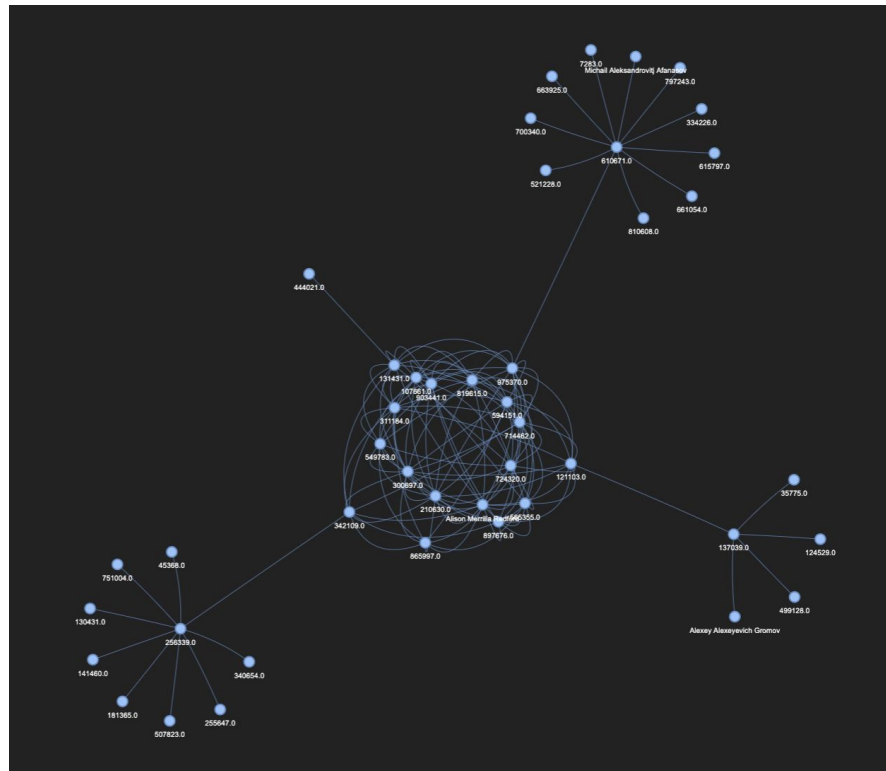
visualization model with a certain name

INPUT :

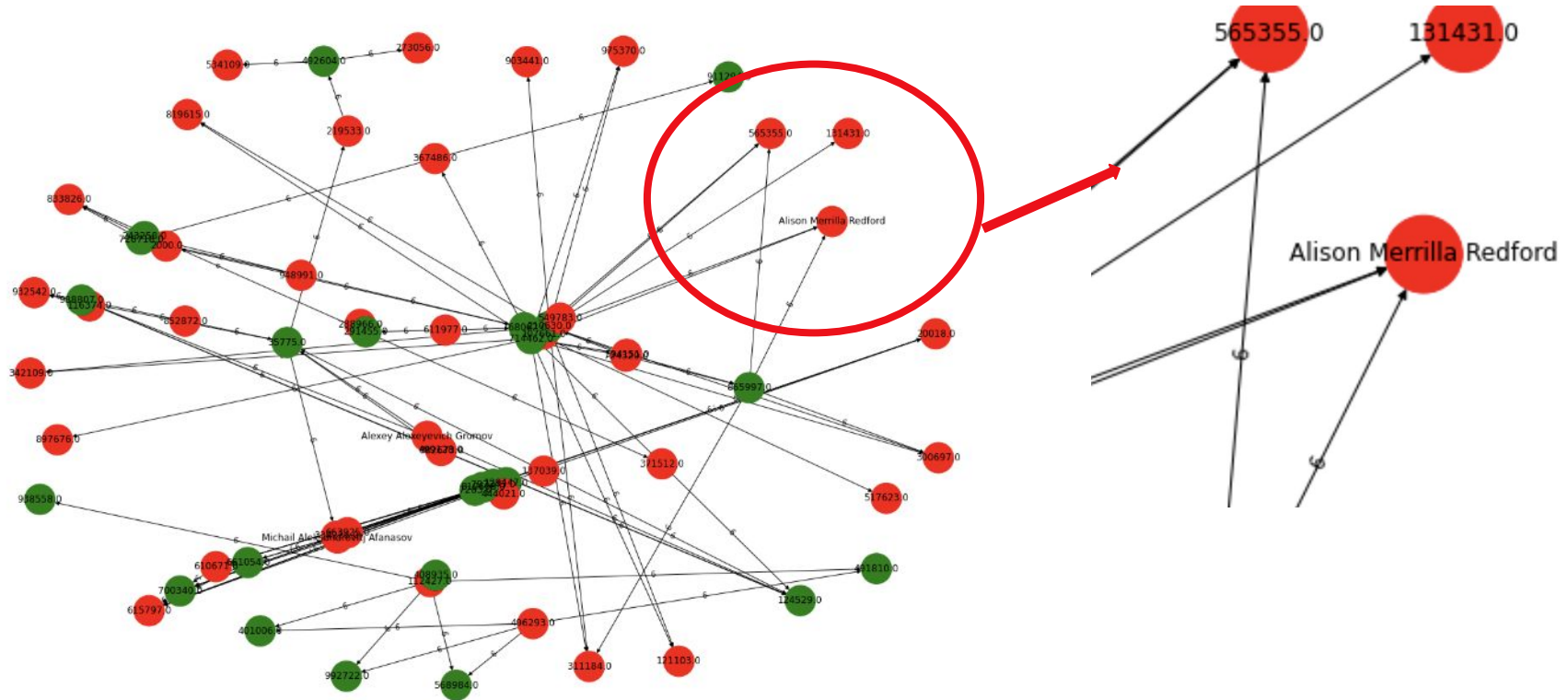
**Alison Merrilla
Redford**



OUTPUT :



Improved Detailed Visualization Model



Reflection & Improvement

- Clients who have transaction with clients in high/medium risk may increase their account risk, we may think about use the network as a new feature
- Money laundering detection is a complex thing, still need model improvement to detect bad actors and related clients, e.g. combine graph model; add more bad actors information to improve the model

Thank you

