MLinAML

Machine Learning for Anti Money Laundering

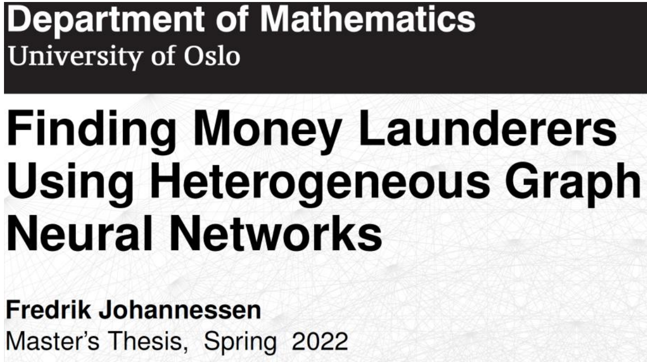
Martin Jullum,

Senior Research Scientist,
Norwegian Computing Center

Oslo Machine Learning Meetup, March 2nd 2023

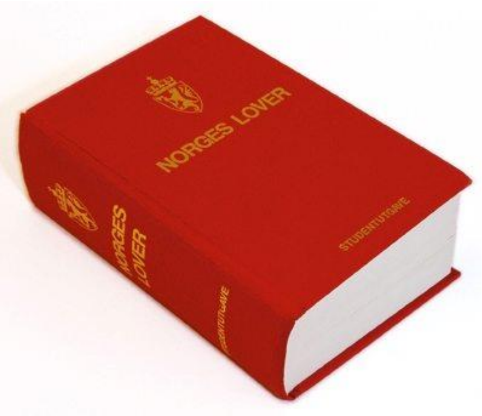
Schedule

1. (Anti) Money Laundering
   - What, why, how
2. ML model for detecting money laundering transactions
   - Method, results and limitations
3. Ongoing work
   a) Algorithms for finding suspicious transaction patterns
   b) GNNs for detecting money launderers
4. Q&A



1. (Anti) Money Laundering

- What, why, how

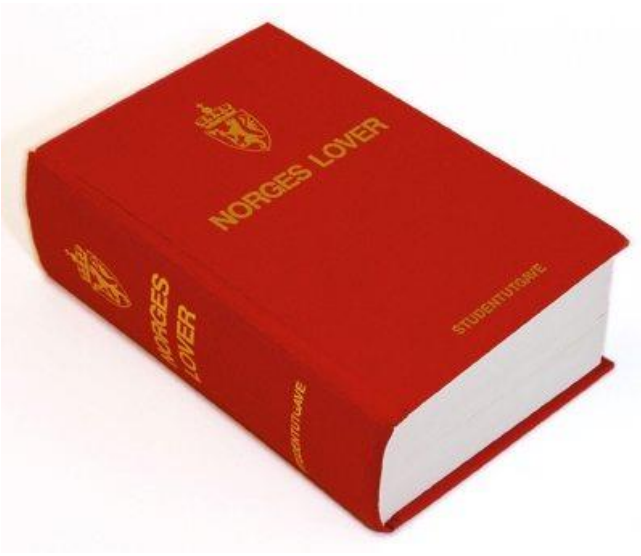


Money laundering

- Making money from criminal activity appear legal
- Examples
  - Buy antics with dirty money – state as attic finding – sell legally
  - Incorporate criminal funds in your own legal business

Money laundering

- Making money from criminal activity appear legal
- Examples
  - Buy antics with dirty money – state as attic finding – sell legally
  - Incorporate criminal funds in your own legal business



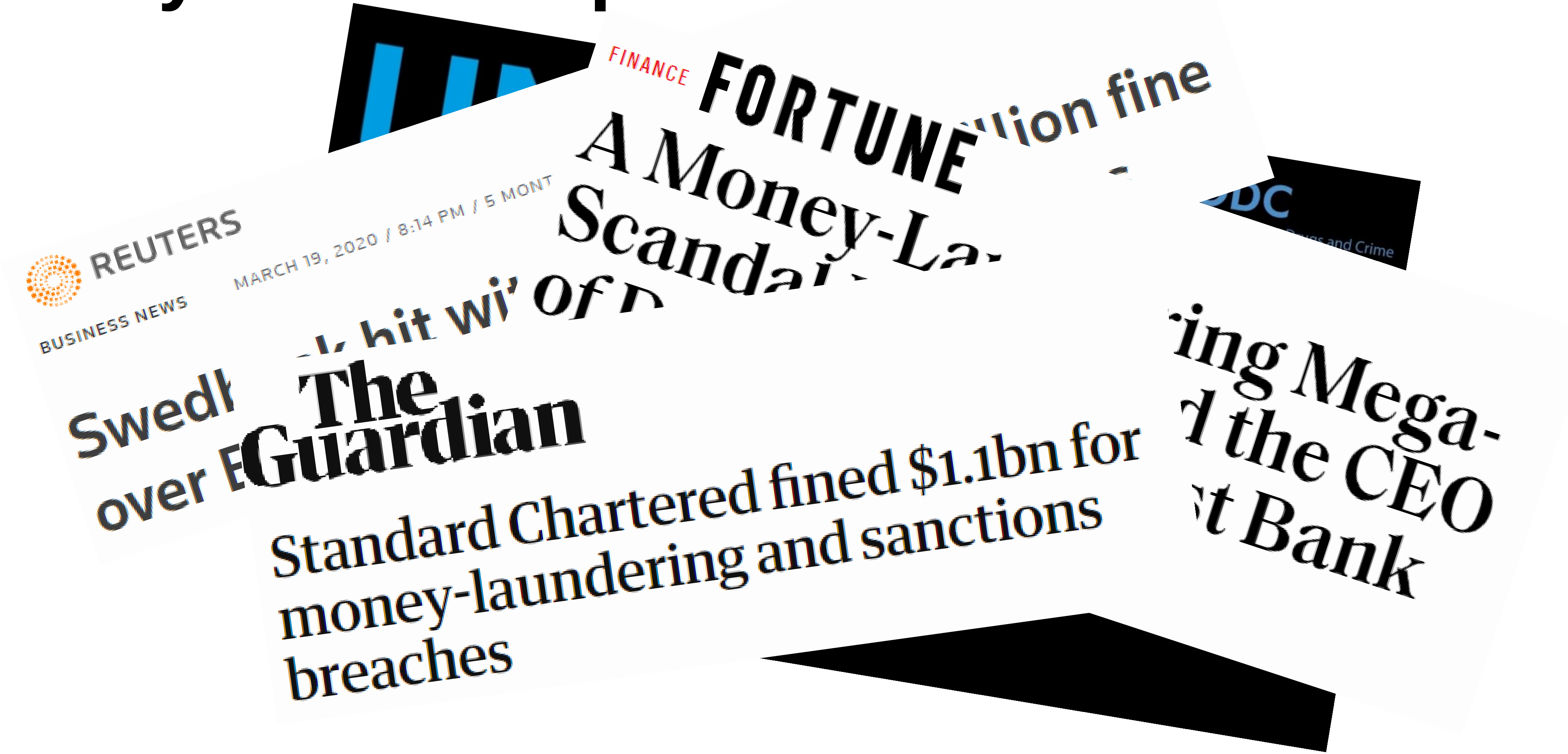
► All financial institutions are legally binded to report “suspicious transactions” to Økokrim

Why is AML important?

Why is AML important?

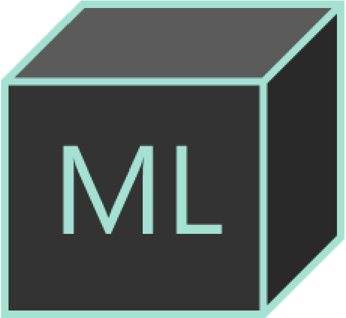
Why is AML important?

Why is AML important?



2. ML model for detecting money laundering transactions

- Method, results and limitations

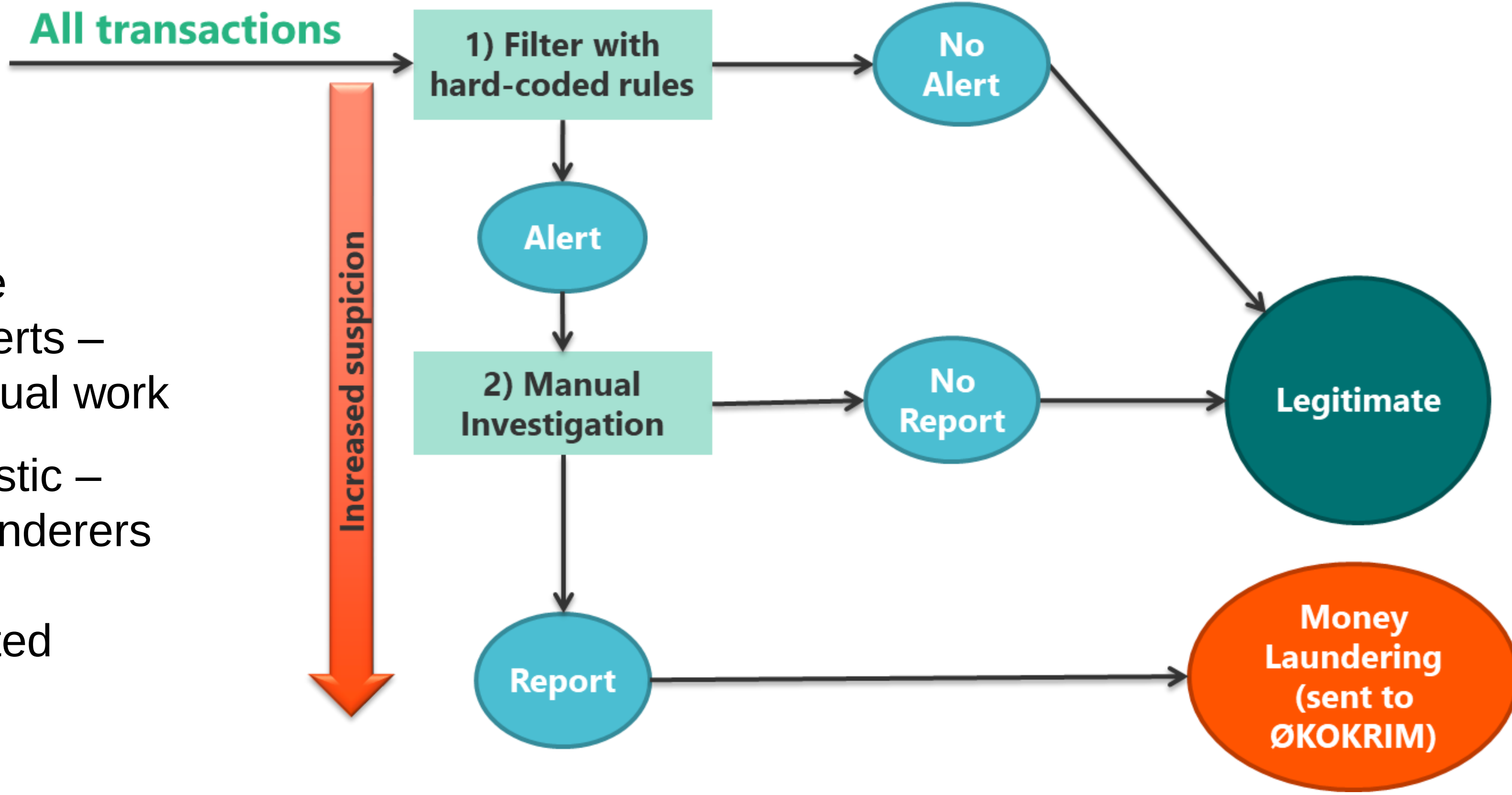


dmlc
XGBoost

Detecting money laundering transactions with machine learning

Martin Jullum, Anders Løland and Ragnar Bang Huseby
Norwegian Computing Center, Oslo, Norway, and
Geir Ånonsen and Johannes Lorentzen
DNB, Oslo, Norway

Current AML process at DNB



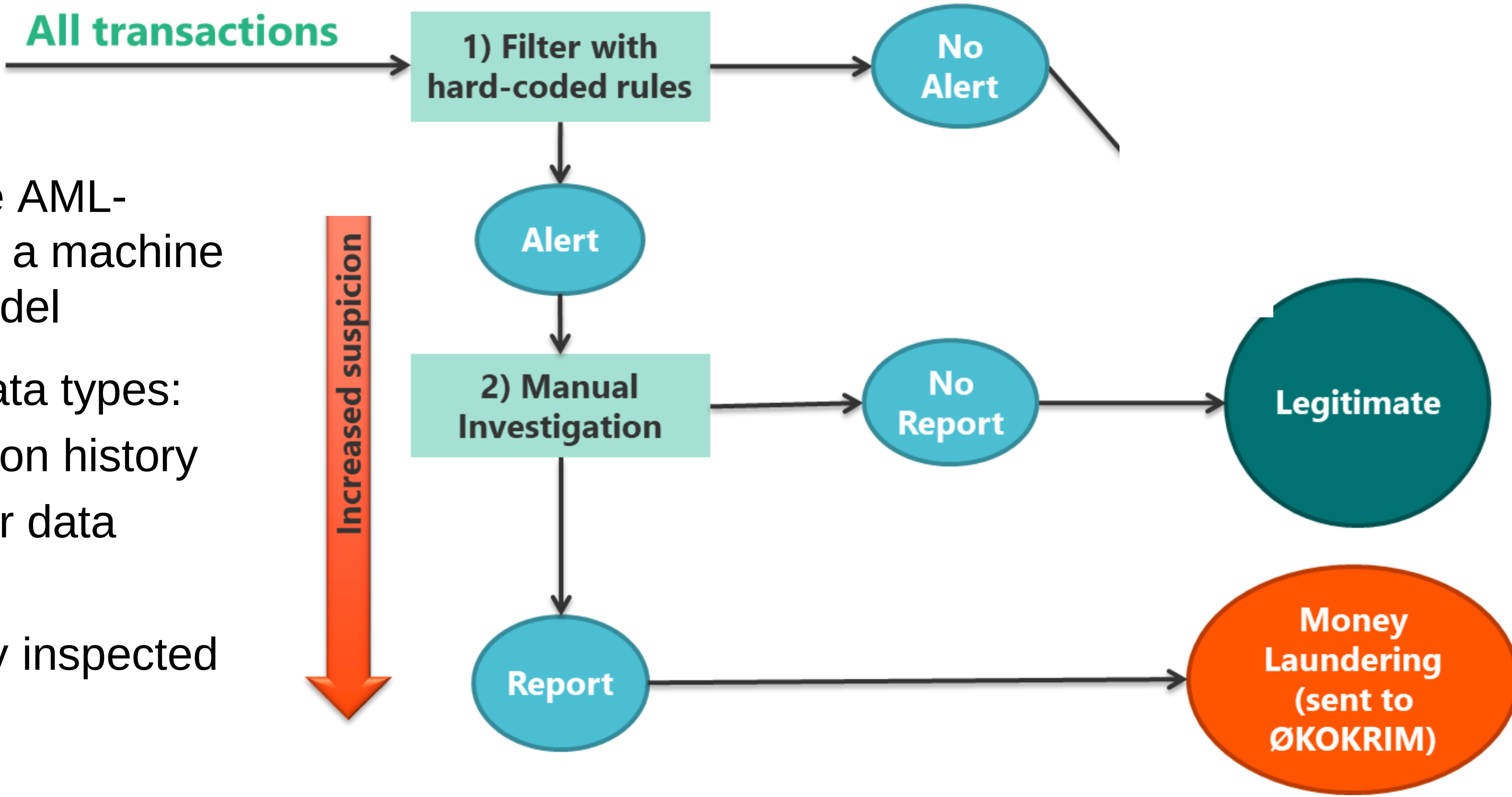
Weaknesses

- Many false positive alerts – much manual work
- Too simplistic – Money launderers are more sophisticated

What we have done

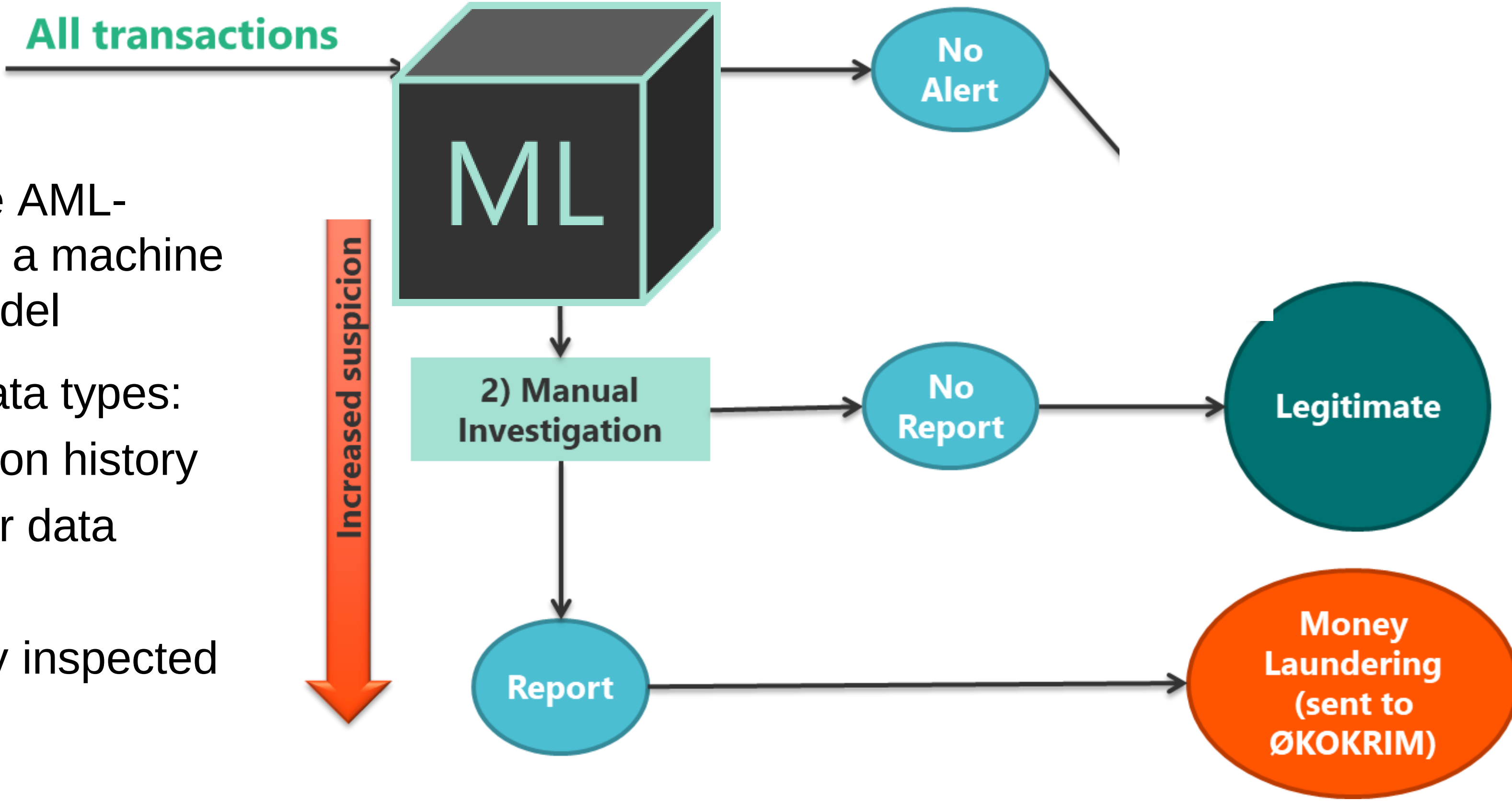
More realistic setting!

- Replace the AML-system with a machine learning model
- Available data types:
  - transaction history
  - customer data
  - alerts
  - manually inspected cases



What we have done

More realistic setting!

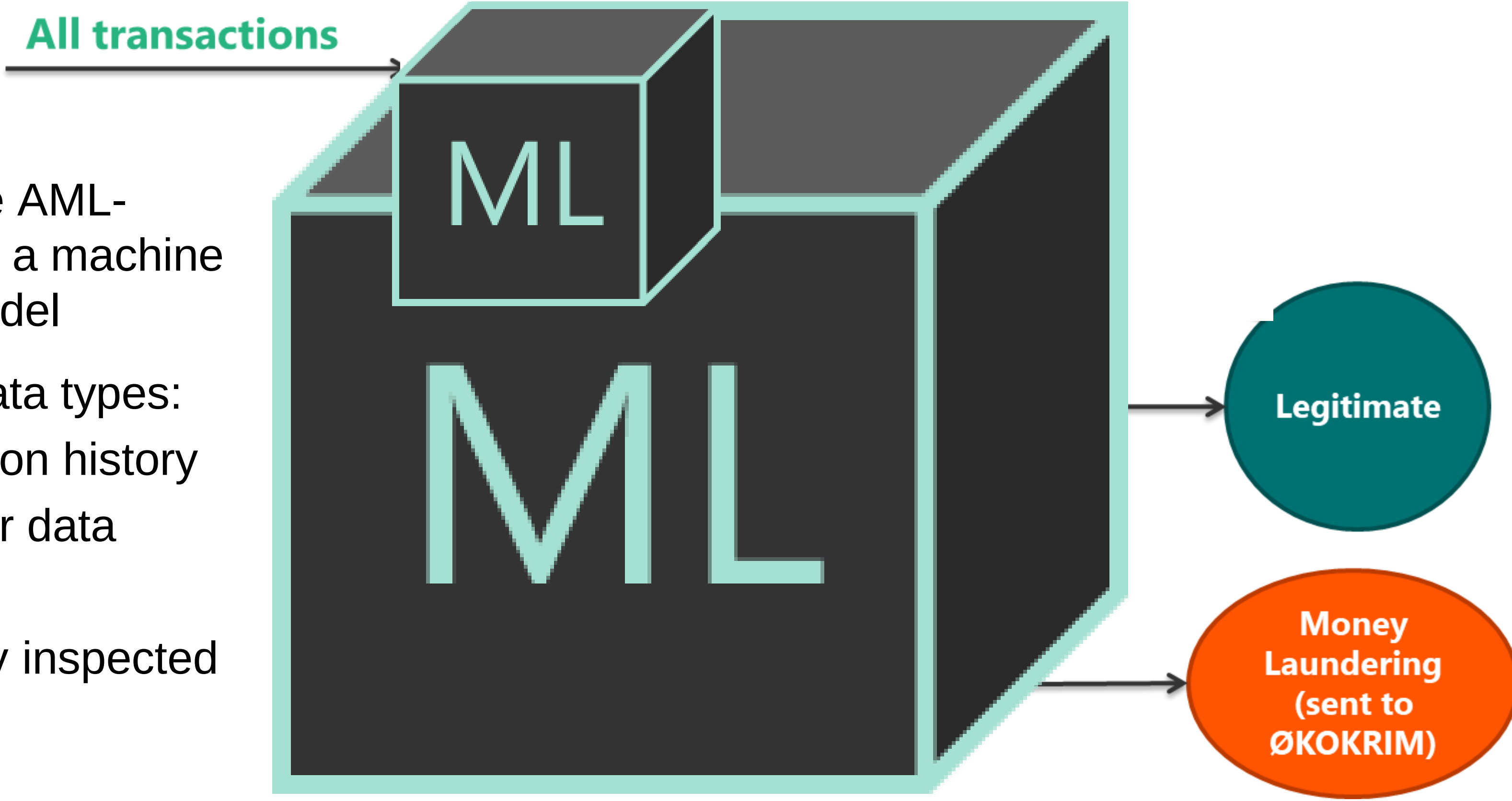


- Replace the AML-system with a machine learning model
- Available data types:
  - transaction history
  - customer data
  - alerts
  - manually inspected cases

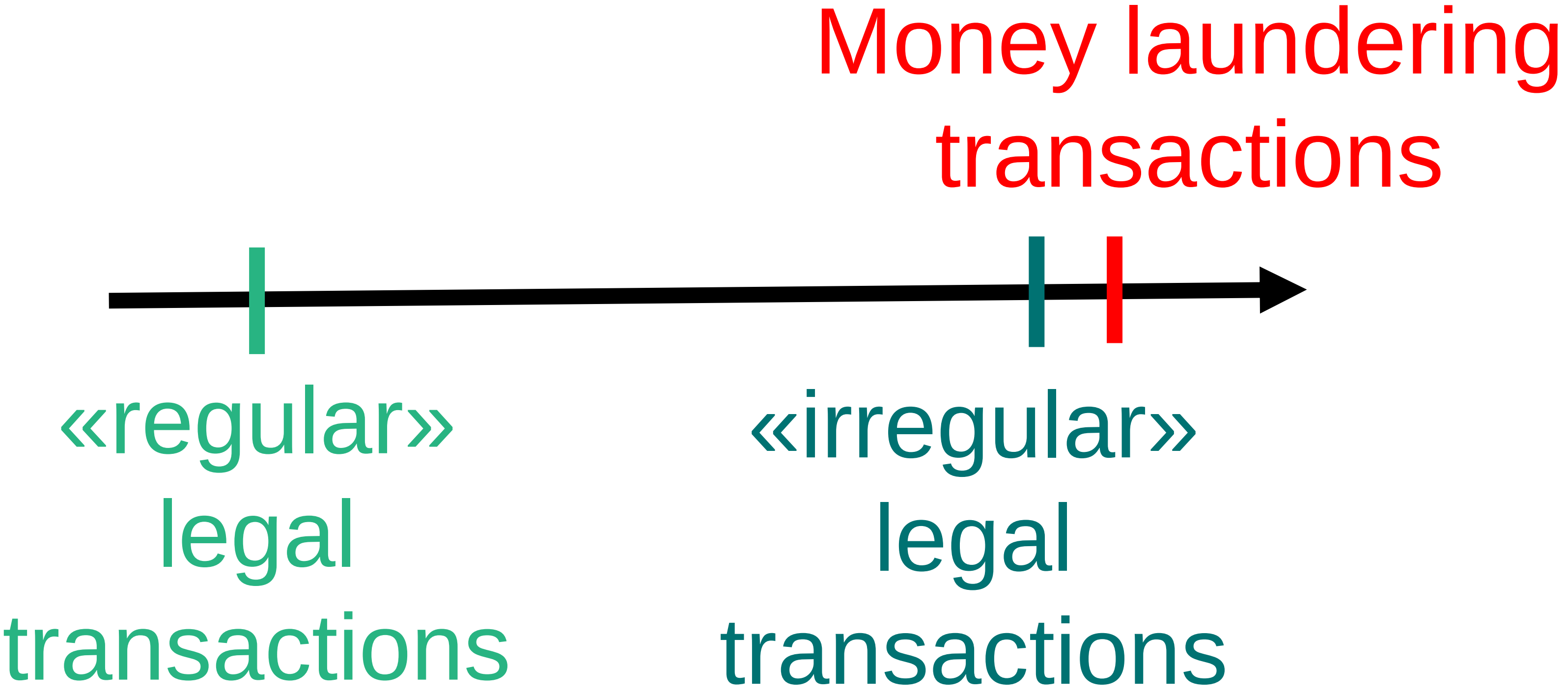
What we have done

More realistic setting!

- Replace the AML-system with a machine learning model
- Available data types:
  - transaction history
  - customer data
  - alerts
  - manually inspected cases



What makes this hard?

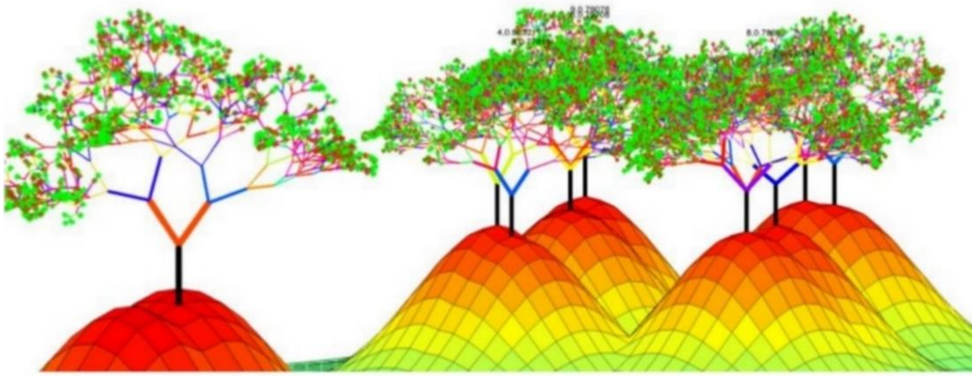


Modelling

- Binary response (Y): Transaction sent to Økokrim (Yes = 1, no = 0)
- Want to model $P(Y = 1|\text{data related to present transaction})$

- State of the art: **Gradient boosting** machines (GBM)
- **XGBoost –** very efficient and flexible implementation of GBM based on **tree models**

dmlc
XGBoost



Transforming raw data (feature engineering)

- XGBoost requires numeric tabular data as input!

Raw input data

- Specific transaction info
- Background info about sender/receiver
- Sender/receiver's transaction history
- Previously reported transactions from sender/receiver

Y	X1	X2	X3	X4	X5	X6
1	0,453406	0,992838	0,734389	0,159918	0,397515	0,949952
0	0,274	0,654207	0,169886	0,493841	0,407112	0,939789
0	0,741897	0,855005	0,585788	0,366456	0,365123	0,57955
1	0,488119	0,465754	0,716517	0,493048	0,855049	0,632114
0	0,134458	0,762057	0,848194	0,098779	0,872603	0,063026
0	0,531914	0,998817	0,808215	0,060721	0,716595	0,35374
0	0,341509	0,8398	0,637808	0,48304	0,279987	0,730286
0	0,530306	0,463271	0,338713	0,986781	0,925251	0,272484
1	0,864123	0,652763	0,689599	0,080937	0,990294	0,364736
0	0,106812	0,900351	0,450224	0,143815	0,593244	0,020764

1716 columns (features)

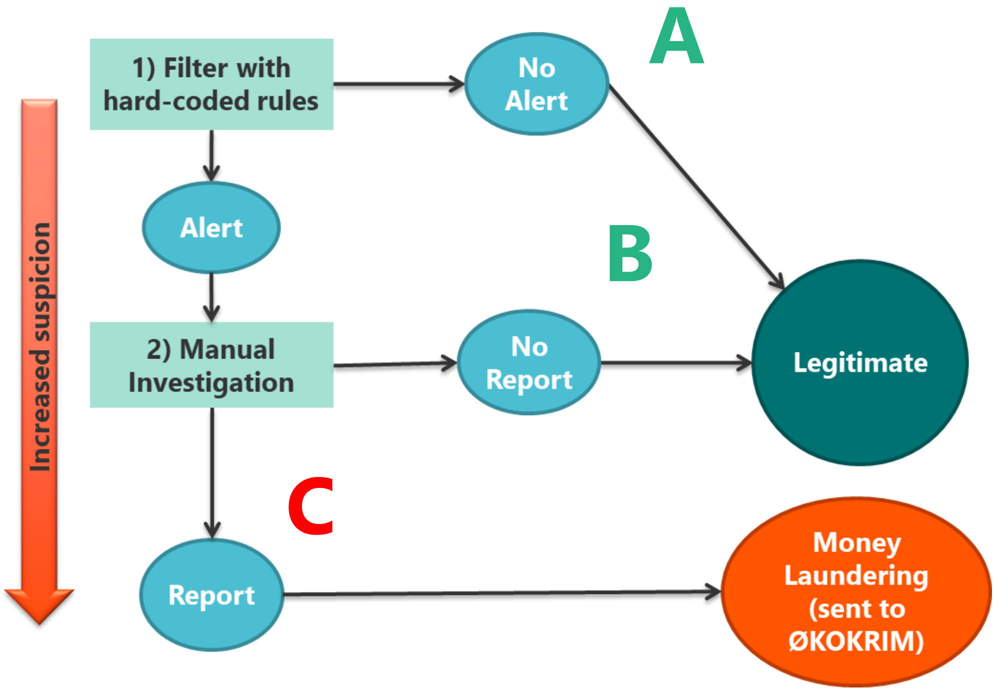
Data refinement

2 years of modellable transaction data

- All transactions leading to
  - A report (C)
  - An alert, but no report (B)
- A sample of normal transactions (A)

Data refinement

- We chose #A = #B
- Use only one transaction from each manual investigation (2)
- No transactions with same sender/receiver two consecutive days

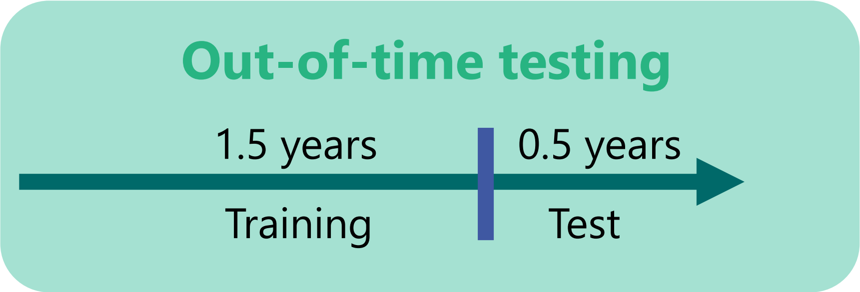


Training, testing and modelling

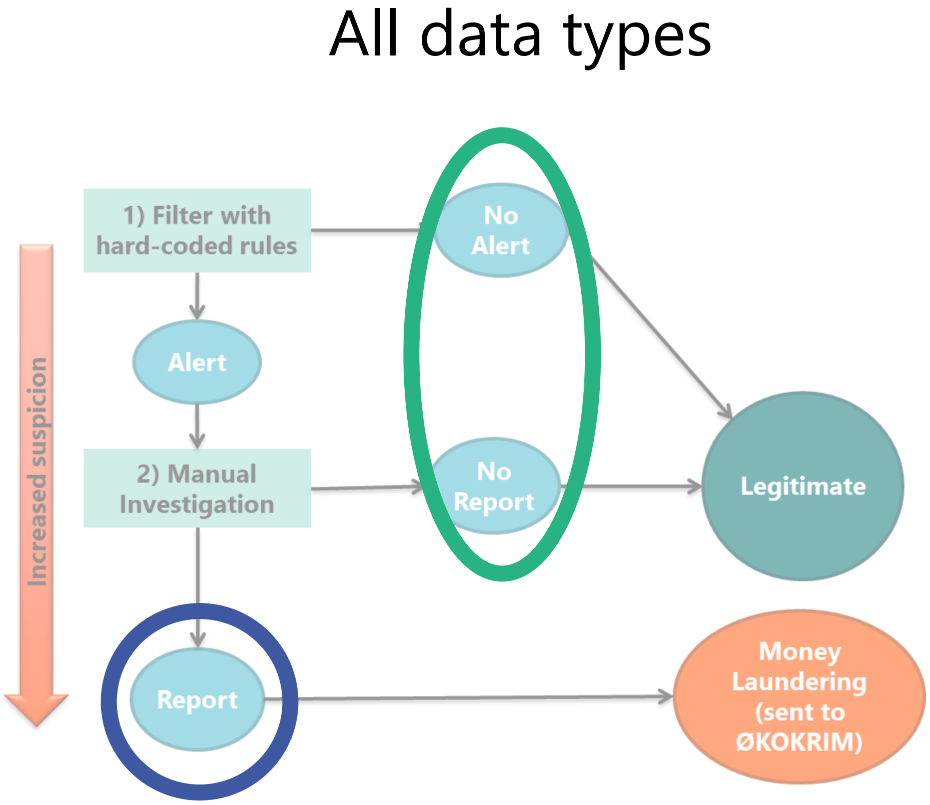
Modelling

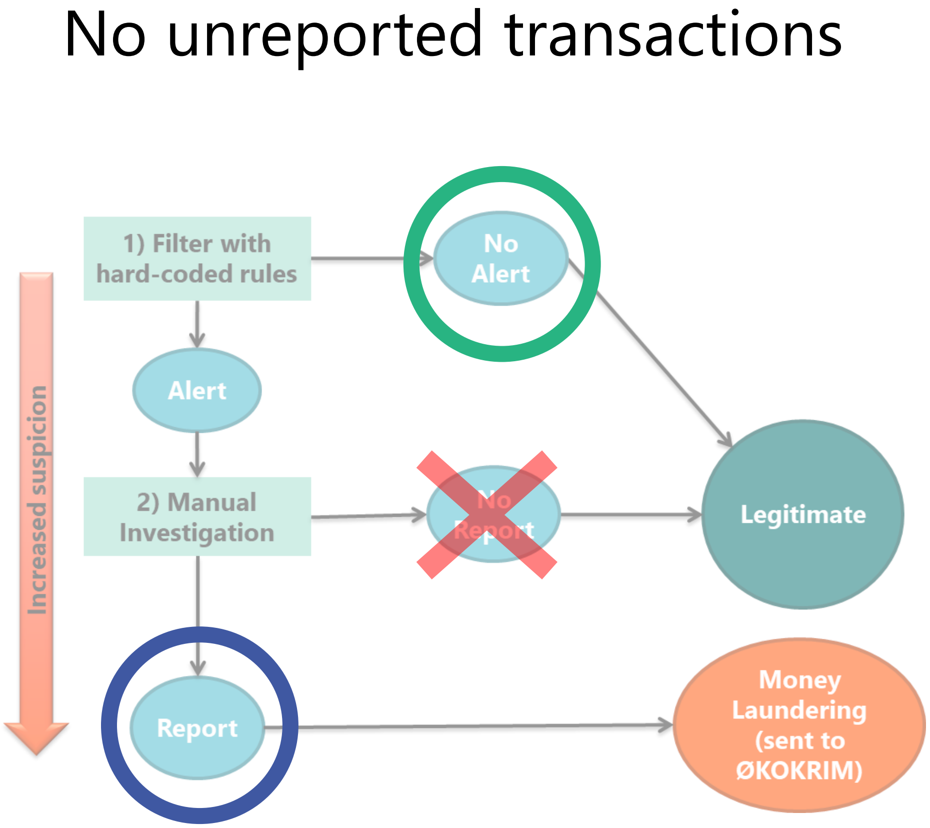
- 10-fold cross validation (CV)
- Stopping criterion (# boosting rounds): AUC
- Tuning: Random + iterative grid-search
- Model trained on GPU
- Final model used for prediction on test data:

$$\hat{f}(x_{\text{test}}) = \frac{1}{10}\sum_{i=1}^{10} \hat{f}_{cv,-i}(x_{\text{test}})$$



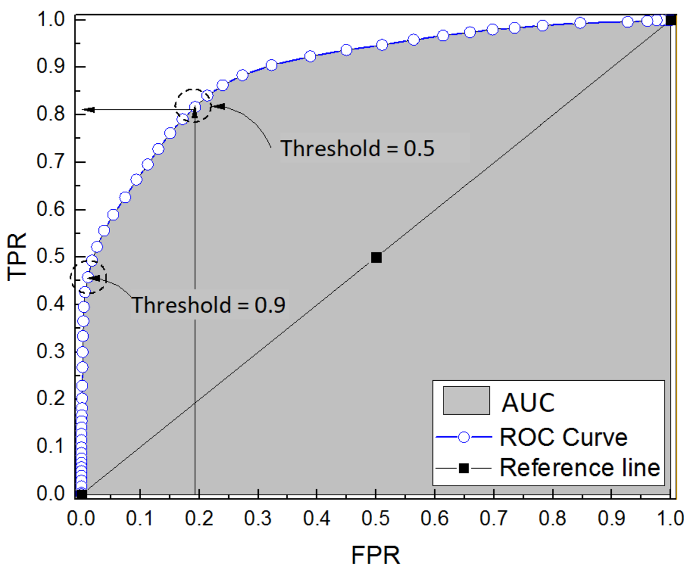
2 training scenarios





Evaluation metrics

Ranking:
AUC



Probabilities:
Brier score

$$\frac{1}{n_{\text{test}}}\sum_{i=1}^{n_{\text{test}}}(y_i - \hat{p}_i)^2$$

Comparing scenarios

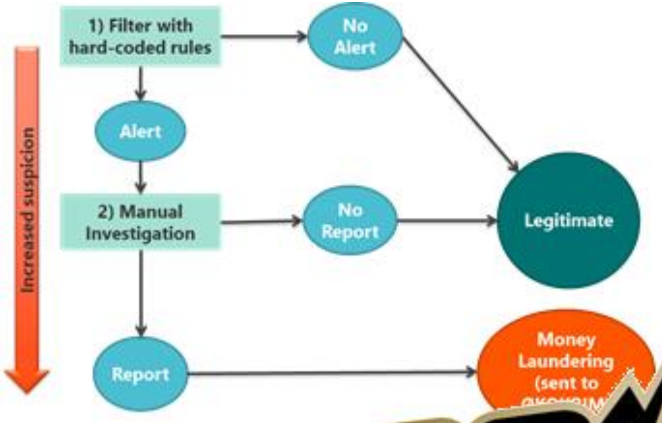
	All data types	No unreported transactions
AUC	**0.907**	**0.852**
Brier	**0.025**	**0.340**

Much better!

ML vs current AML system

- Hard to properly compare

- **PPP = Proportion of Positive Predictions:** Proportion of transactions that needs to be controlled to find 95% of the reported transactions



	ML (all data types)	Current system
PPP	31.5 %	48.9 %

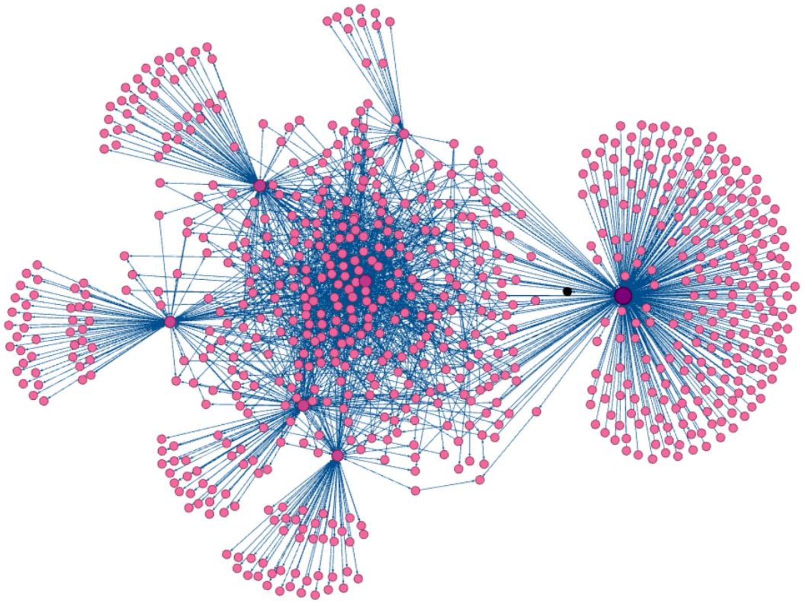
Limitations

- We are not really using the **time-evolving transaction network**
  - **Who** are you sending/receiving money to/from
  - **When** are you sending/receiving
- Social/professional network information is not used
- Many variables – complicates putting the model into production
- The model only learns what has already been reported

3. Ongoing work

a) Algorithms for finding suspicious transaction patterns

b) GNNs for detecting money launderers



3. Ongoing work

a) Algorithms for finding suspicious transaction patterns

b) GNNs for detecting money launderers

Previous slide

Attempts to address

a)

b)

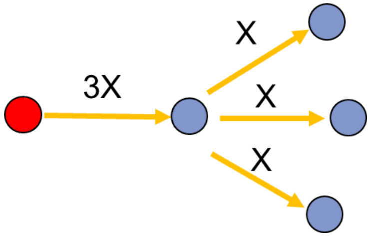
Limitations

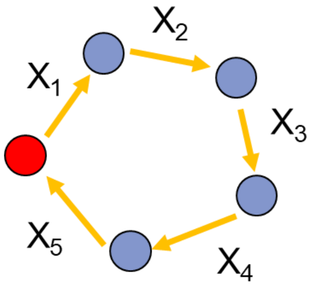
- We are not really using the **time-evolving transaction network**
  - **Who** are you sending/receiving money to/from
  - **When** are you sending/receiving
- Social/professional network information is not used
- Many variables – complicates putting the model into production
- The model only learns what has already been reported

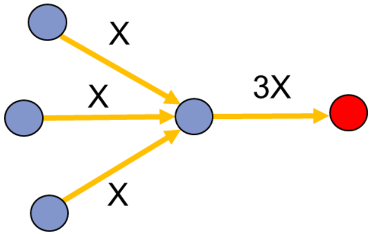
Algorithms for finding suspicious transaction patterns

Attempts to detect unknown money laundering cases

- We have developed algorithms that searches for typical/hypothetical money laundering patterns
  - Time aspect is central: Fast in – fast out
  - Specific transaction types
- Problematic that we only see part of the transaction network
- Still promising initial results

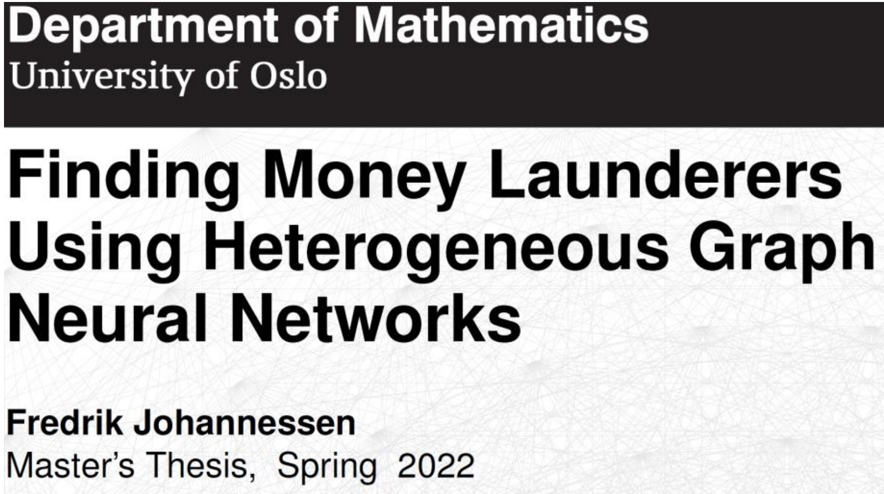






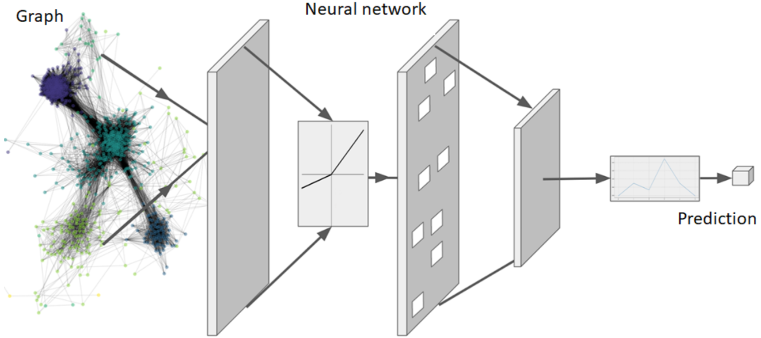
GNNs for detecting money launderers

- Work initiated as part of master thesis
- In the process of writing a paper
- Graph Neural Network
  - Class of methods for building predictive models directly on graph data


Department of Mathematics
University of Oslo

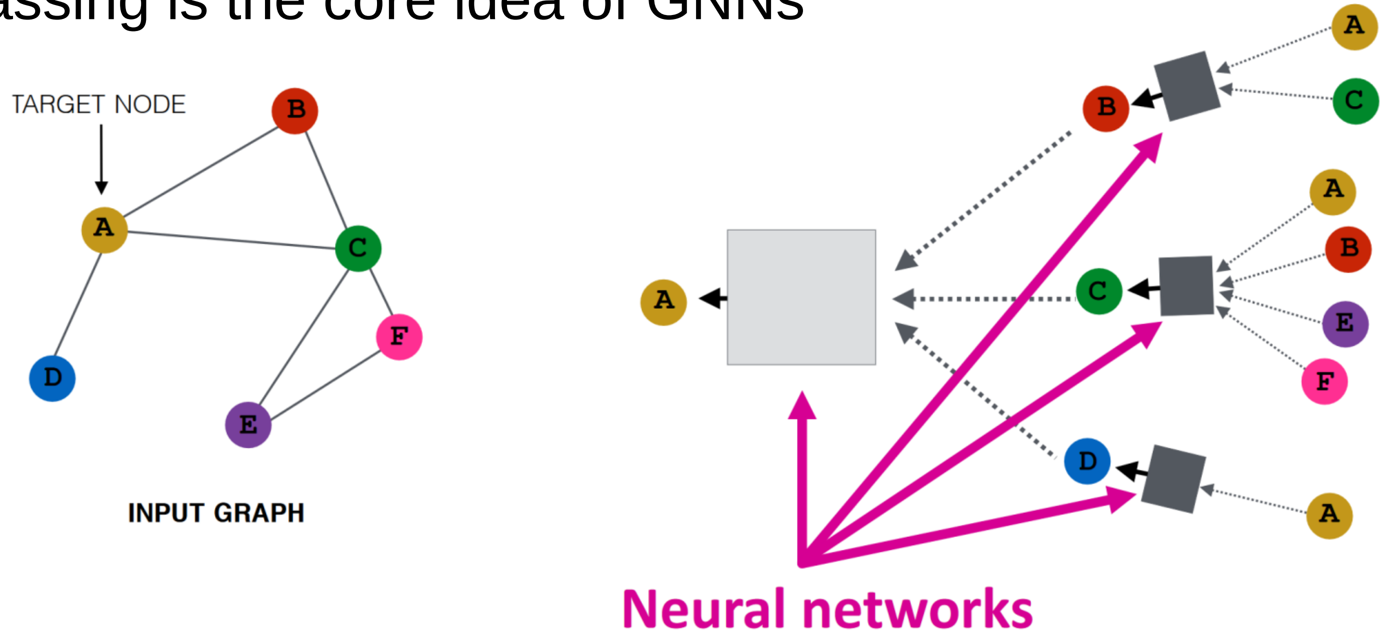
Finding Money Launderers Using Heterogeneous Graph Neural Networks

Fredrik Johannessen
Master's Thesis, Spring 2022



GNNs for detecting money launderers

- Message passing is the core idea of GNNs

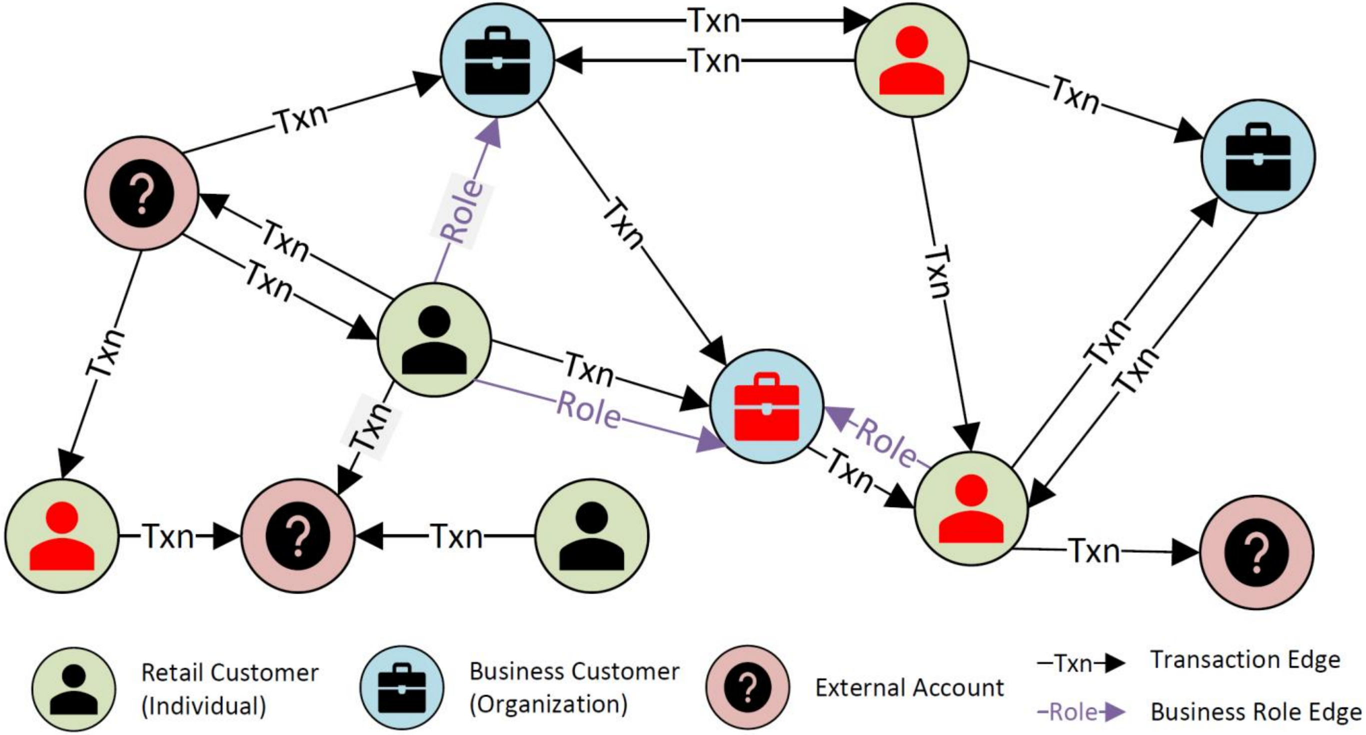


- Aggregation parameters are shared across nodes – allowing generalizing to new nodes

GNNs for detecting money launderers

Our graph

- Heterogeneous in both edges and nodes
  - Transaction + role network
  - Individual + organization + external accounts

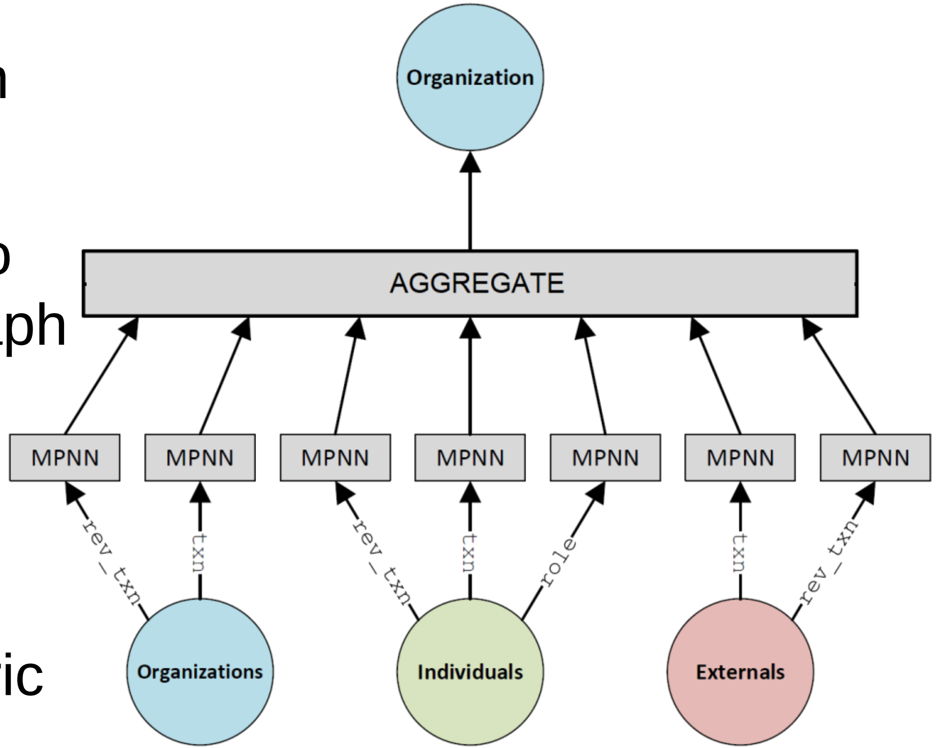


GNNs for detecting money launderers

- MPNN
  - Homogeneous GNN that can handle edge features
- We expand the MPNN model to work on our heterogeneous graph
  - Separate MPNN-models for each combination of

$$\mathtt{node\_type} \xrightarrow{\mathtt{edge\_type}} \mathtt{node\_type}$$

- All built within Pytorch Geometric
- Good results!



4. Q&A

Thank you!

Martin Jullum – martinjullum.com – jullum@nr.no