Graph Neural Networks for Real World *Fraud Detection*

Feng Zhao & Tingting Qiao

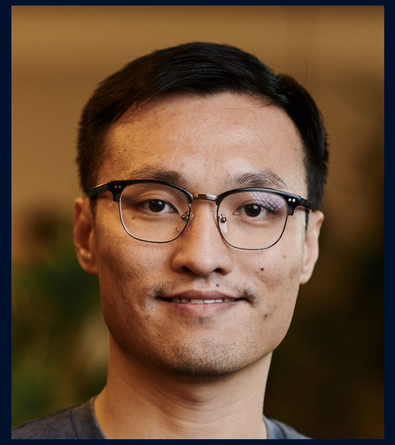
#PyData Amsterdam 2023

adyen

PyData

engineered
for ambition

About us



Feng Zhao

- Data scientist @Adyen
- Ph.D. in computer science
- Applied ML, GNNs, deep learning
- LinkedIn: @ZhaoFeng
- Email: feng.zhao@adyen.com

Tingting Qiao

- Data scientist @Adyen
- Ph.D in computer science
- Deep learning, NLP, generative models
- LinkedIn: TingtingQiao
- Email: tingting.qiao@adyen.com

adyen

engineered
for ambition

Takeaways

- ***Challenging*** business problem
- ***Amazingly performing*** GNN-based solution!
- ***How*** did we put GNNs in production?
- All lessons learned, ***which we normally don't share XD***
- Q&A

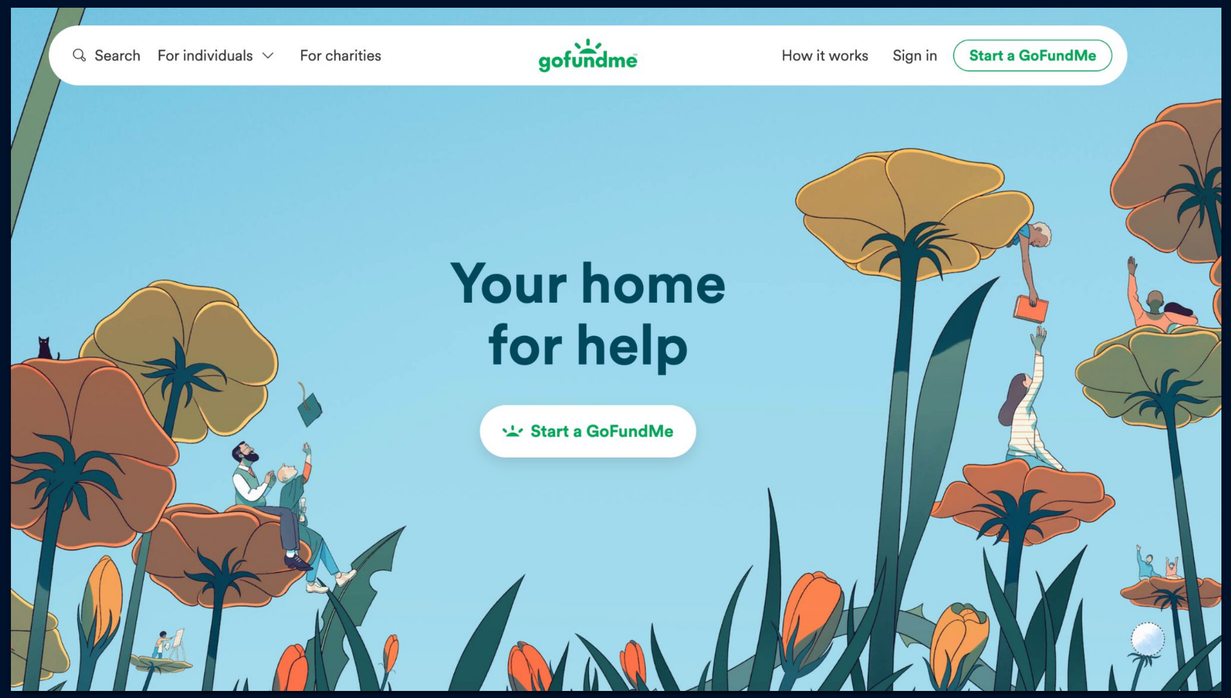
adyen

engineered
for ambition

What’s our business problem?

Business problem

One example - GoFundMe



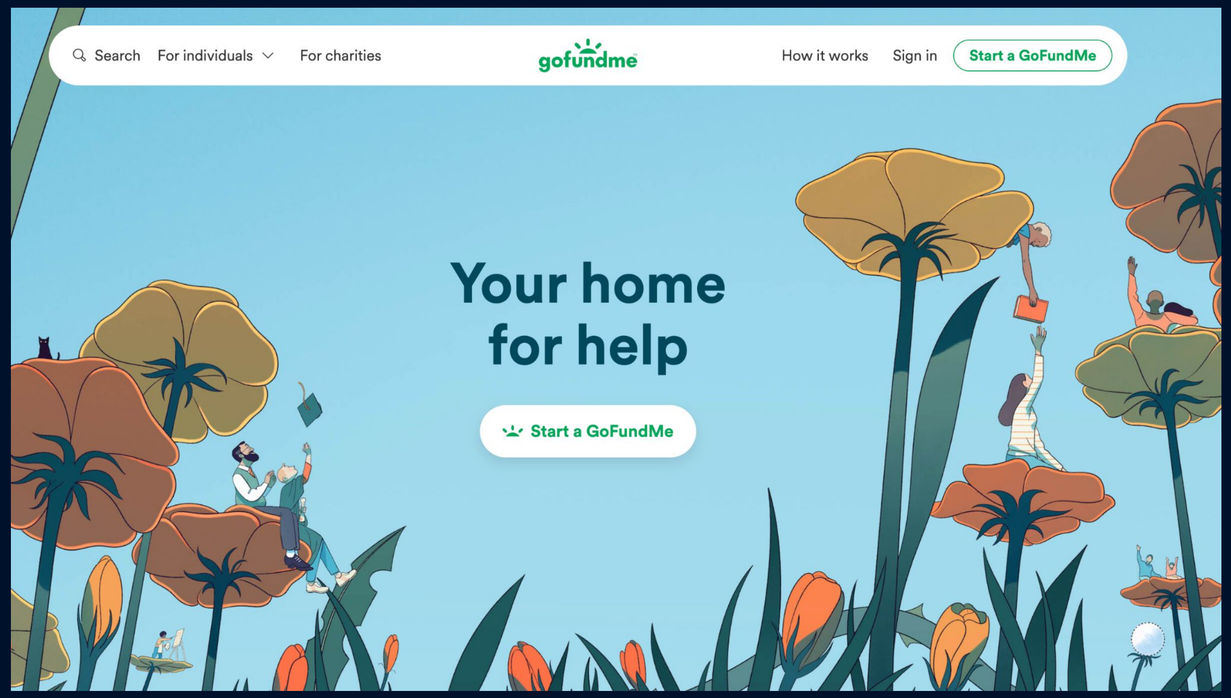
- Crowdfunding platform
- Users can open an account to raise funds

adyen

engineered
for ambition

Business problem

One example - GoFundMe



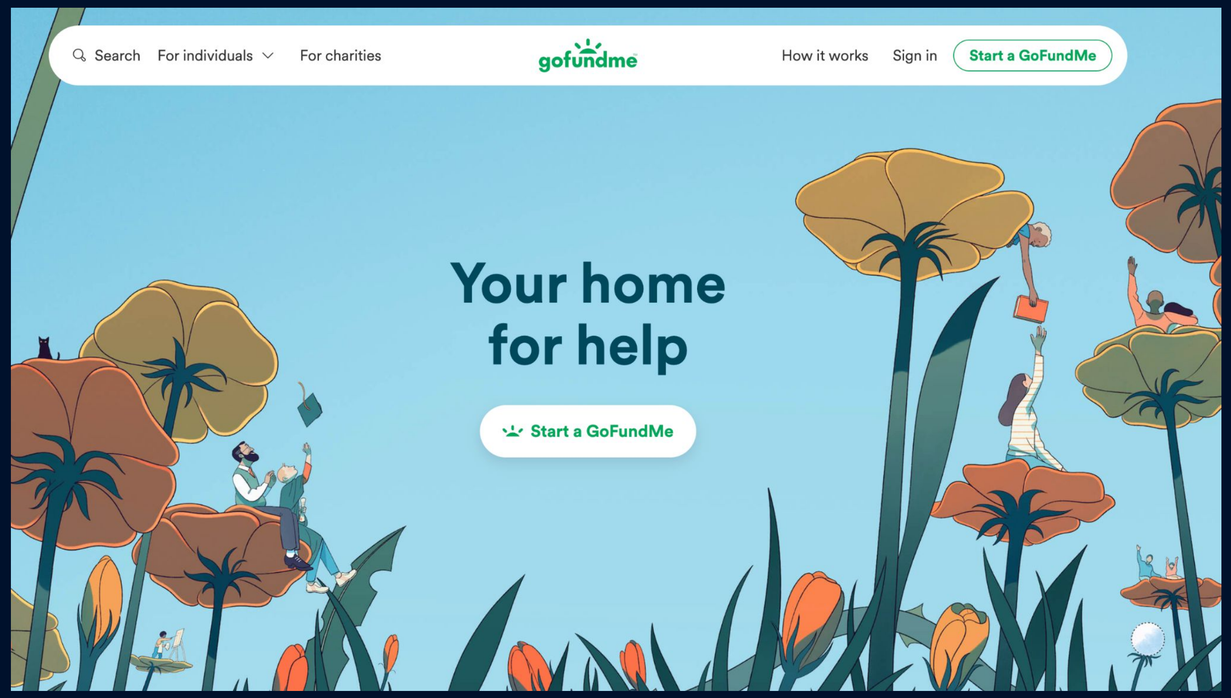
- Crowdfunding platform
- Users can open an account to raise funds
- **What if a fraudster opens an account there?**

adyen

engineered for ambition

Business problem

One example - GoFundMe



- Crowdfunding platform
- Users can open an account to raise funds
- What if a fraudster opens an account?
  - Money laundry
  - Cash out a stolen credit card
  - Chargeback & refund
  - ....

adyen

engineered for ambition

Score @Adyen

Fraud detection for platforms and marketplaces

- External product:
  - Safeguard our merchants from seller frauds
- Internal tool:
  - Fight against financial losses

adyen

engineered for ambition

Amsterdam · August 31, 2021

Adyen launches Score with GoFundMe — a machine learning tool to easily identify malicious platform users

Adyen is first-to-market with a machine learning driven solution, on a single platform, for signaling irregular activity and monitoring platform compliance.

Download the image above to use for your publication. Download image ›

Adyen, the global payments platform of choice for many of the world's leading companies, today launched Score. The company is first-to-market with a machine learning driven solution, on a single platform, for signaling irregular activity and monitoring platform compliance. By leveraging data insights analyzing the platform merchant's data and flagging unusual platform user behavior, Score helps Adyen's merchants prevent misuse of the platform. As Score provides insights via a broad set of risk signals, the feature is of significant support for platform merchants' compliance procedures. By improving effectiveness and reducing time spent on platform user security reviews, Score increases operational scalability for platforms.

Score @Adyen

Fraud detection for platforms and marketplaces

- Money Laundering
- Cash out stolen credit card
- Chargeback & refund fraud
- Account takeover
- Front store
- Identity theft
- and more … 😱

adyen

engineered
for ambition

Business formulation

Problem	Fraud detection
Scope	Sellers on platforms and marketplaces
Solution	Supervised learning, binary classification model → GNNs
Input	Payment data / Transactions
Output	Binary prediction
Goal	○ Maximize true positives ○ Keep false positives at a low level

adyen

engineered
for ambition

Read Adyen's way of managing account holder risk

GNNs based solution

Why GNNs?

Seller 1

Seller 2

Seller 3

Seller 4

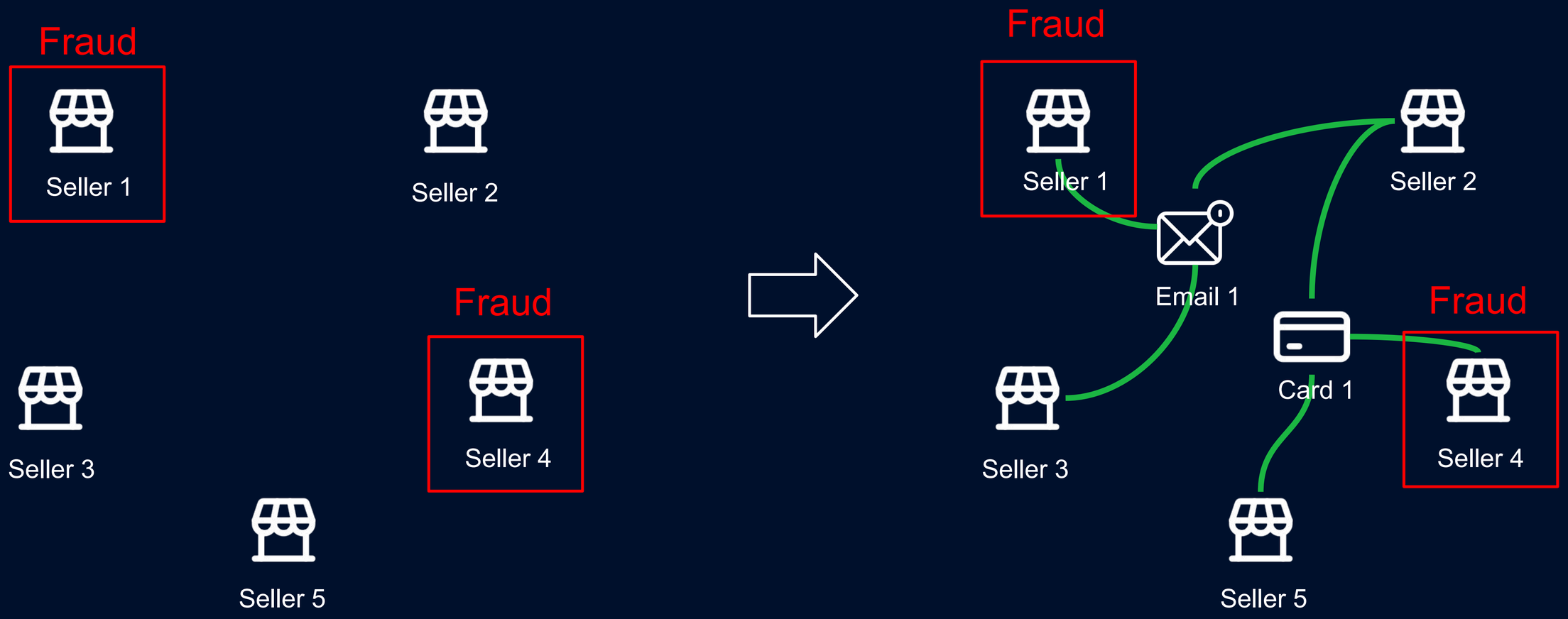
Seller 5

- **Business rules**
- **Traditional ML**
- **Can we do better?**

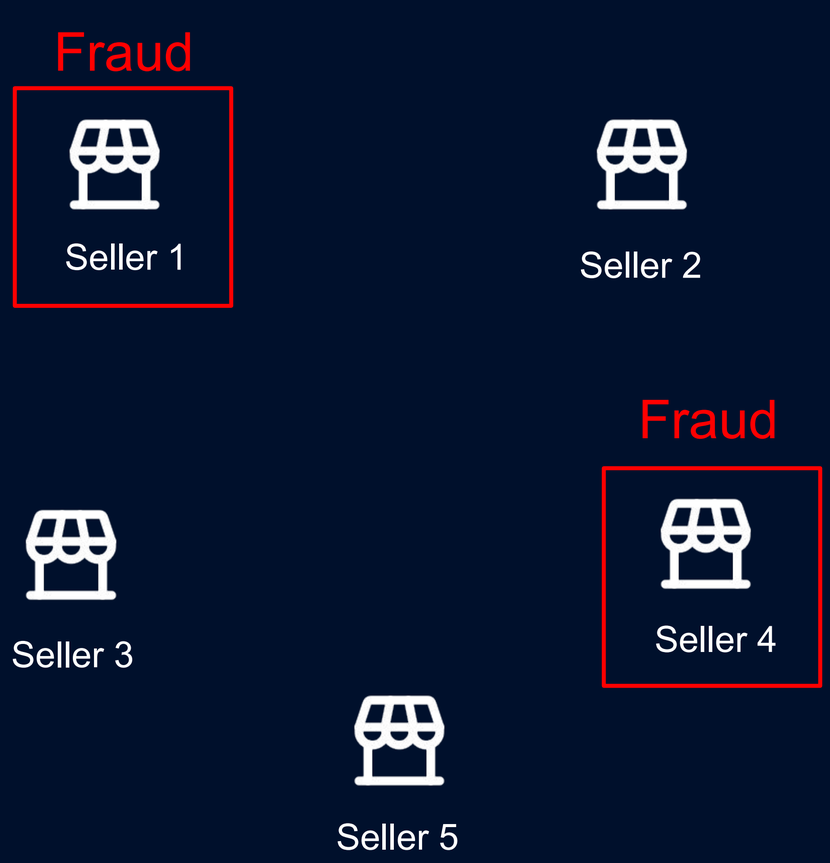
Why GNNs?

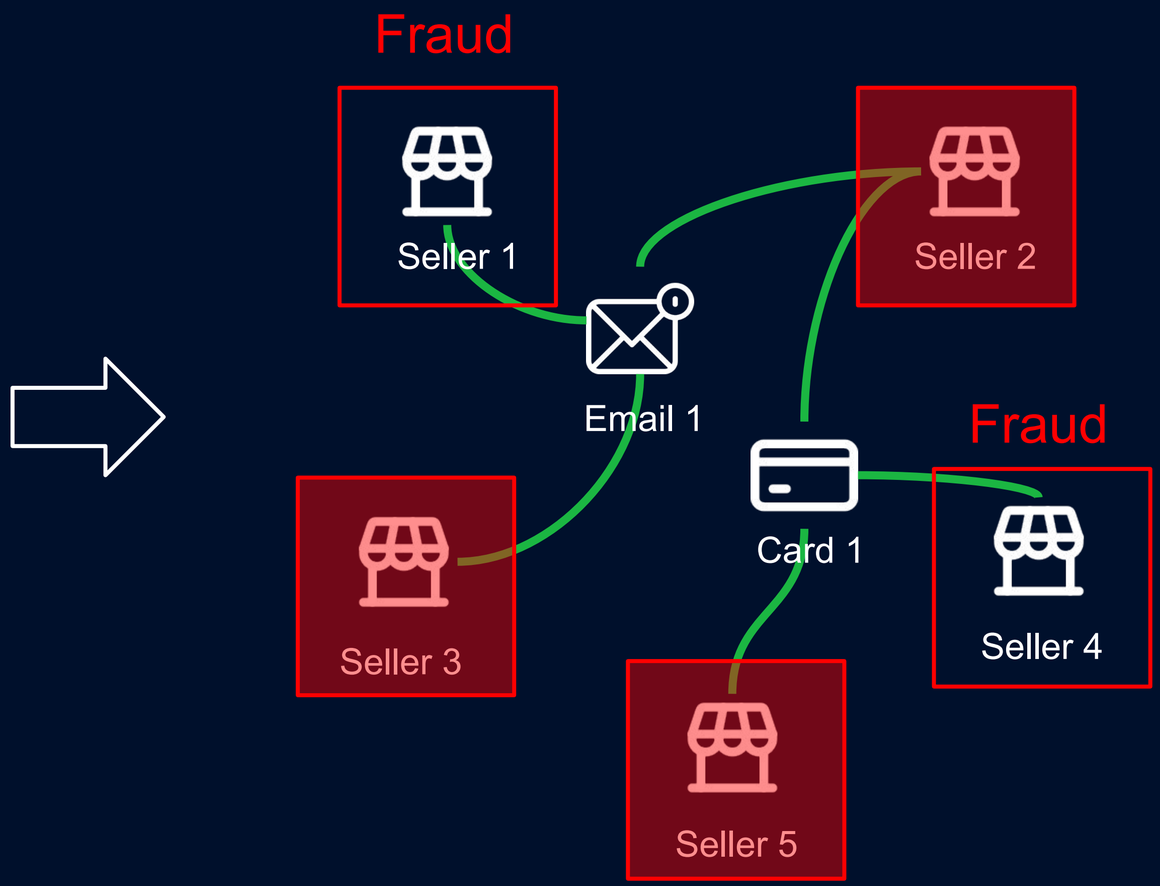

Seller 1
Seller 2
Seller 3
Seller 4
Seller 5
Seller 1
Seller 2
Email 1
Card 1
Seller 3
Seller 4
Seller 5

Why GNNs?


Fraud
Seller 1
Seller 2
Fraud
Seller 4
Seller 3
Seller 5
Fraud
Seller 1
Seller 2
Email 1
Fraud
Card 1
Seller 4
Seller 3
Seller 5

Why GNNs?


Fraud
Seller 1
Seller 2
Fraud
Seller 4
Seller 3
Seller 5


Fraud
Seller 1
Seller 2
Email 1
Fraud
Card 1
Seller 3
Seller 4
Seller 5

Graph creation

DataFrame → GraphFrame

Transaction table

Trx	Seller	Shopper Card	Shopper Email
1	seller1	card1	email1
2	seller2	card2	email1

Transaction table visualization

DataFrame → GraphFrame

Transaction table

Trx	Seller	Shopper Card	Shopper Email
1	seller1	card1	email1
2	seller2	card2	email1

Transaction table visualization


```
vertex_measures = [Measures.TX_COUNT]
edge_measures = [Measures.TX_COUNT]
graph = (
    transactions
    .src_by(
        SrcVertexColumns.SELLER,
        *vertex_measures
    )
    .dst_by(
        DstVertexColumns.CARD,
        DstVertexColumns.EMAIL,
        *vertex_measures
    )
    .edge_agg(*edge_measures)
    .graph()
)
```

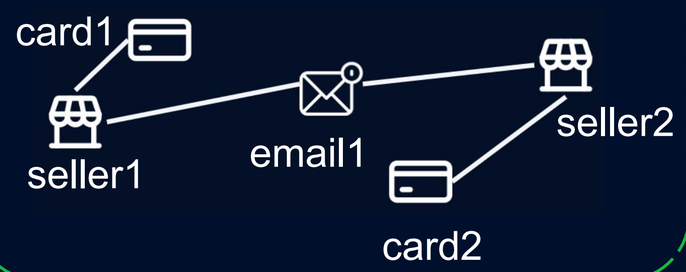
(Code is from our own internal Graph API)

DataFrame → GraphFrame

Transaction table

Trx	Seller	Shopper Card	Shopper Email
1	seller1	card1	email1
2	seller2	card2	email1

Transaction table visualization




```
vertex_measures = [Measures.TX_COUNT]
edge_measures = [Measures.TX_COUNT]
graph = (
    transactions
    .src_by(
        SrcVertexColumns.SELLER,
        *vertex_measures
    )
    .dst_by(
        DstVertexColumns.CARD,
        DstVertexColumns.EMAIL,
        *vertex_measures
    )
    .edge_agg(*edge_measures)
    .graph()
)
```

(Code is from our own internal Graph API)

Vertex table

ID	Type
seller1	Seller
seller2	Seller
card1	Card
card2	Card
email1	Email

Edge table

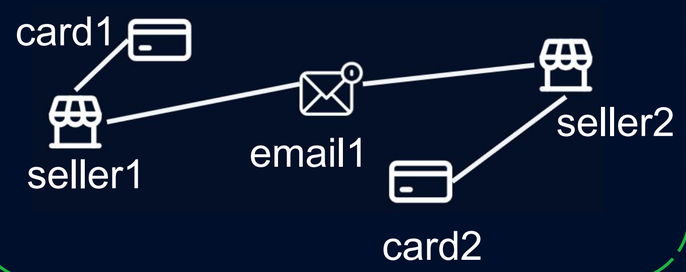
Src	Dst
seller1	card1
seller1	email1
seller2	card2
seller2	email1

DataFrame → GraphFrame

Transaction table

Trx	Seller	Shopper Card	Shopper Email
1	seller1	card1	email1
2	seller2	card2	email1

Transaction table visualization




```
vertex_measures = [Measures.TX_COUNT]
edge_measures = [Measures.TX_COUNT]
graph = (
    transactions
    .src_by(
        SrcVertexColumns.SELLER,
        *vertex_measures
    )
    .dst_by(
        DstVertexColumns.CARD,
        DstVertexColumns.EMAIL,
        *vertex_measures
    )
    .edge_agg(*edge_measures)
    .graph()
)
```

(Code is from our own internal Graph API)

Vertex table

ID	Type	trx_cnt
seller1	Seller	1
seller2	Seller	1
card1	Card	1
card2	Card	1
email1	Email	2

Edge table

Src	Dst	trx_cnt
seller1	card1	1
seller1	email1	1
seller2	card2	1
seller2	email1	1

GraphFrame | Bipartite projection

Heterogeneous graph

1. Focus on seller nodes
2. Reduce the graph size

Seller projected graph

GraphFrame | Bipartite projection

Vertices table

ID	Type	trx_cnt
seller1	Seller	1
seller2	Seller	1
card1	Card	1
card2	Card	1
email1	Email	2

Edge table

Src	Dst	trx_cnt
seller1	card1	1
seller1	email1	1
seller2	card2	1
seller2	email1	1

```
projected_graph = graph.bipartite_projection(
    measures=[num_distinct_shopper_email_id()],
    is_src_projection=True
)
```

Vertex table

ID	Type	trx_cnt
seller1	Seller	1
seller2	Seller	1

Edge table

Src	Dst	email_cnt
seller1	seller2	1

What’s the impact?

Heterogeneous graph

Table	Size
Vertex table	~ 18M
Edge table	~ 22M

Seller projected graph

Table	Size
Vertex table	~ 1M
Edge table	~ 10M

adyen

engineered
for ambition

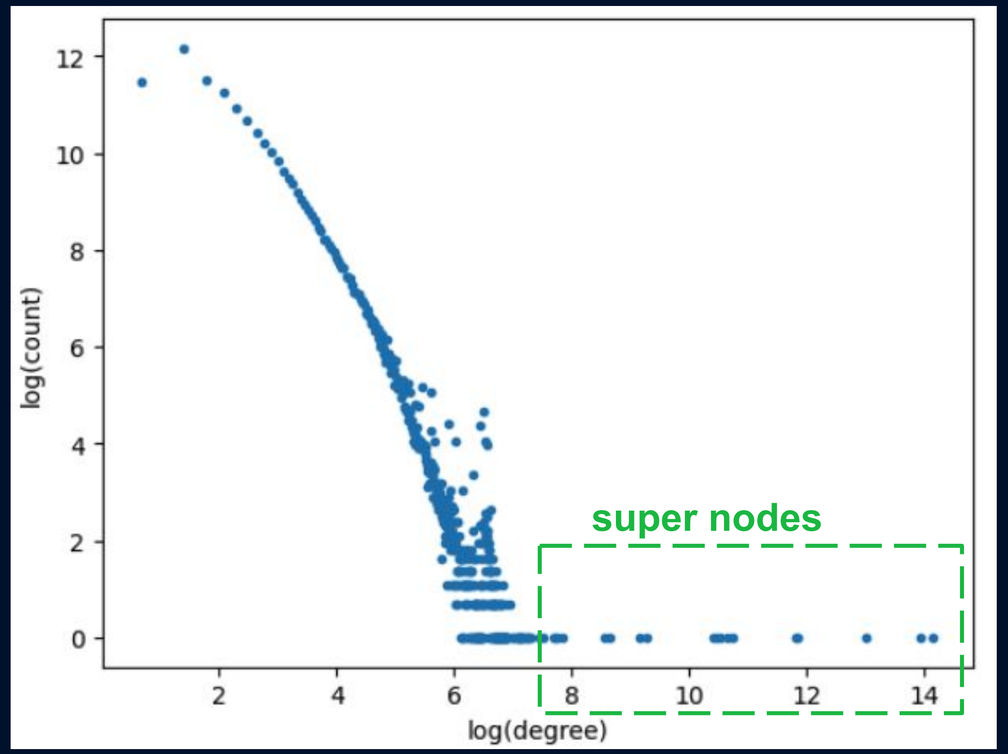
What’s the impact?

Bipartite graph

Table	Size
Vertex table	~ 18M
Edge table	~ 22M

Seller projected graph

Table	Size
Vertex table	~ 1M
Edge table	~ 10M



Degree distribution of seller projected graph

adyen

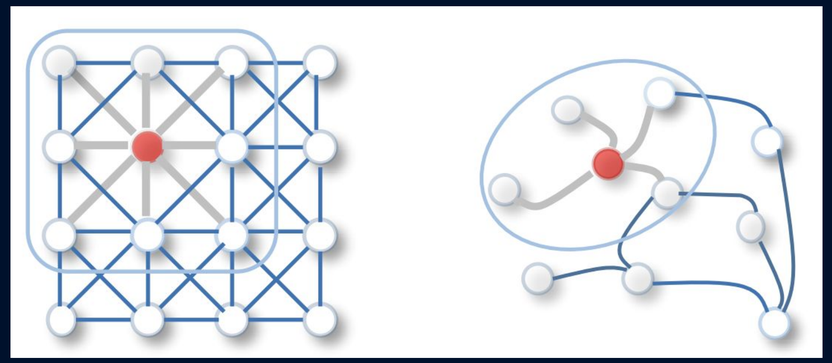
engineered
for ambition

Modeling

Model | GNN design

```python
class GCN(nn.Module):
    new *
    def __init__(self, in_feats, h_feats, num_classes):
        super(GCN, self).__init__()
        self.conv1 = GraphConv(in_feats, h_feats)
        self.conv2 = GraphConv(h_feats, num_classes)

    new *
    def forward(self, g, in_feat):
        h = self.conv1(g, in_feat)
        h = F.relu(h)
        h = self.conv2(g, h)
        return h
```



2D Convolution

Graph Convolution

adyen

engineered
for ambition

★ Figure is from paper <A Comprehensive Survey on Graph Neural Networks>

Model | GNN design

```python
class GCN(nn.Module):
    new *
    def __init__(self, in_feats, h_feats, num_classes):
        super(GCN, self).__init__()
        self.conv1 = GraphConv(in_feats, h_feats)
        self.conv2 = GraphConv(h_feats, num_classes)

    new *
    def forward(self, g, in_feat):
        h = self.conv1(g, in_feat)
        h = F.relu(h)
        h = self.conv2(g, h)
        return h
```

Takeaways:

1. There no best models, only happy model tuning
2. Start with a baseline model, e.g. GCN
3. Try with different Graph Conv layers, e.g. SageConv
4. Nice reading <Graph Representation Learning>

adyen

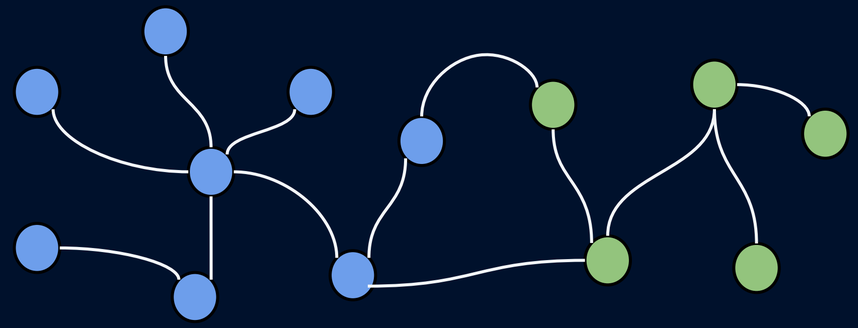
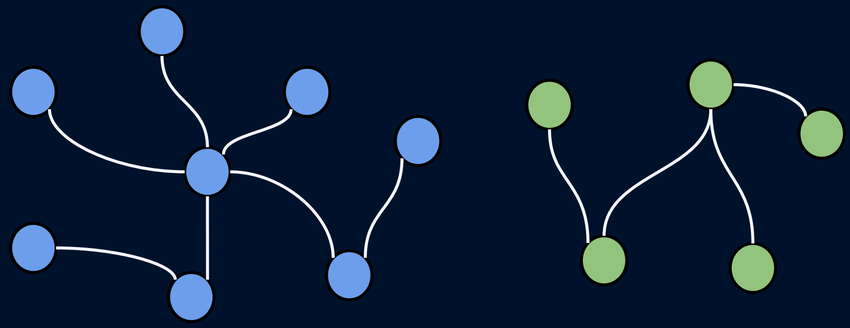
engineered
for ambition

Model | ML setups

	Details	Why?
Graph type	Seller projected graph	• Significantly reduced graph size • Speed up training
Problem	Node classification	We treat each of the sellers as one node in the graph
Training parameters	Weights of positive samples	Imbalanced dataset
Deep Learning framework	DGL	• Well maintained • Good documentation • Support popular algorithms

Model | Training approach

Transductive or Inductive?

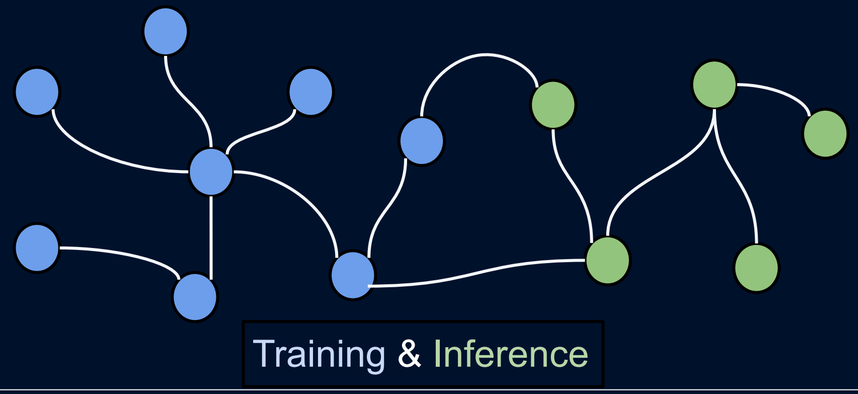
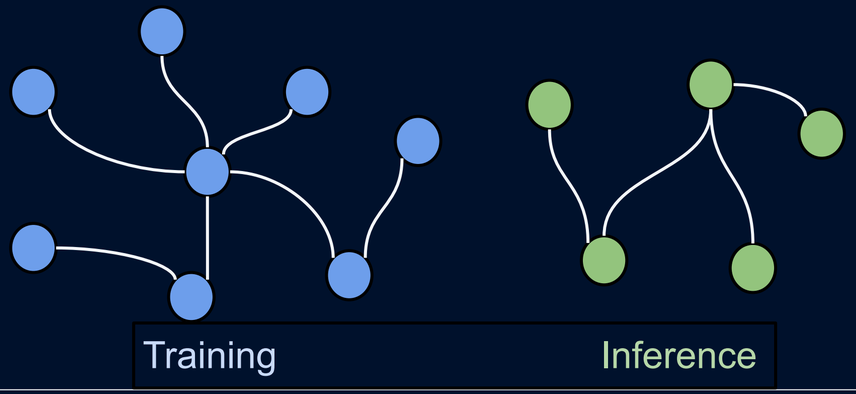
Aspect	Transductive Learning	Inductive Learning (our choice)
Illustration	 Training & Inference	 Training Inference

adyen

engineered
for ambition

Model | Training approach

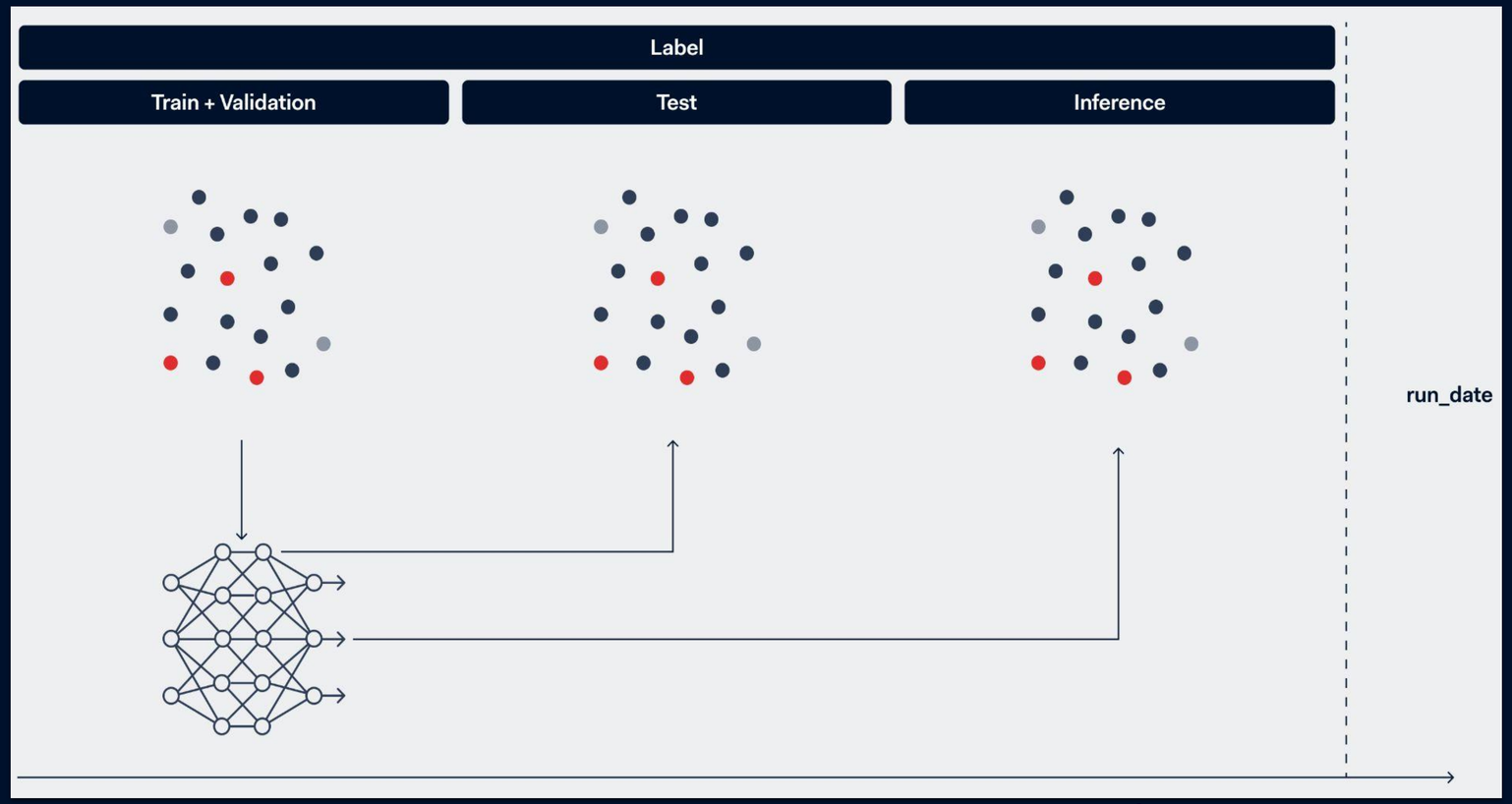
Transductive or Inductive?

Aspect	Transductive Learning	Inductive Learning (our choice)
Illustration		
Data Split	Fixed graph for train/test/inference	Different train/test/inference graphs
Generalization	Limited to the observed graph	Generalizes to unseen nodes
Scalability	More suitable for small to medium-sized graphs	Suitable for large graphs (by splitting it into 3 graphs)

adyen

engineered
for ambition

Model | Training approach



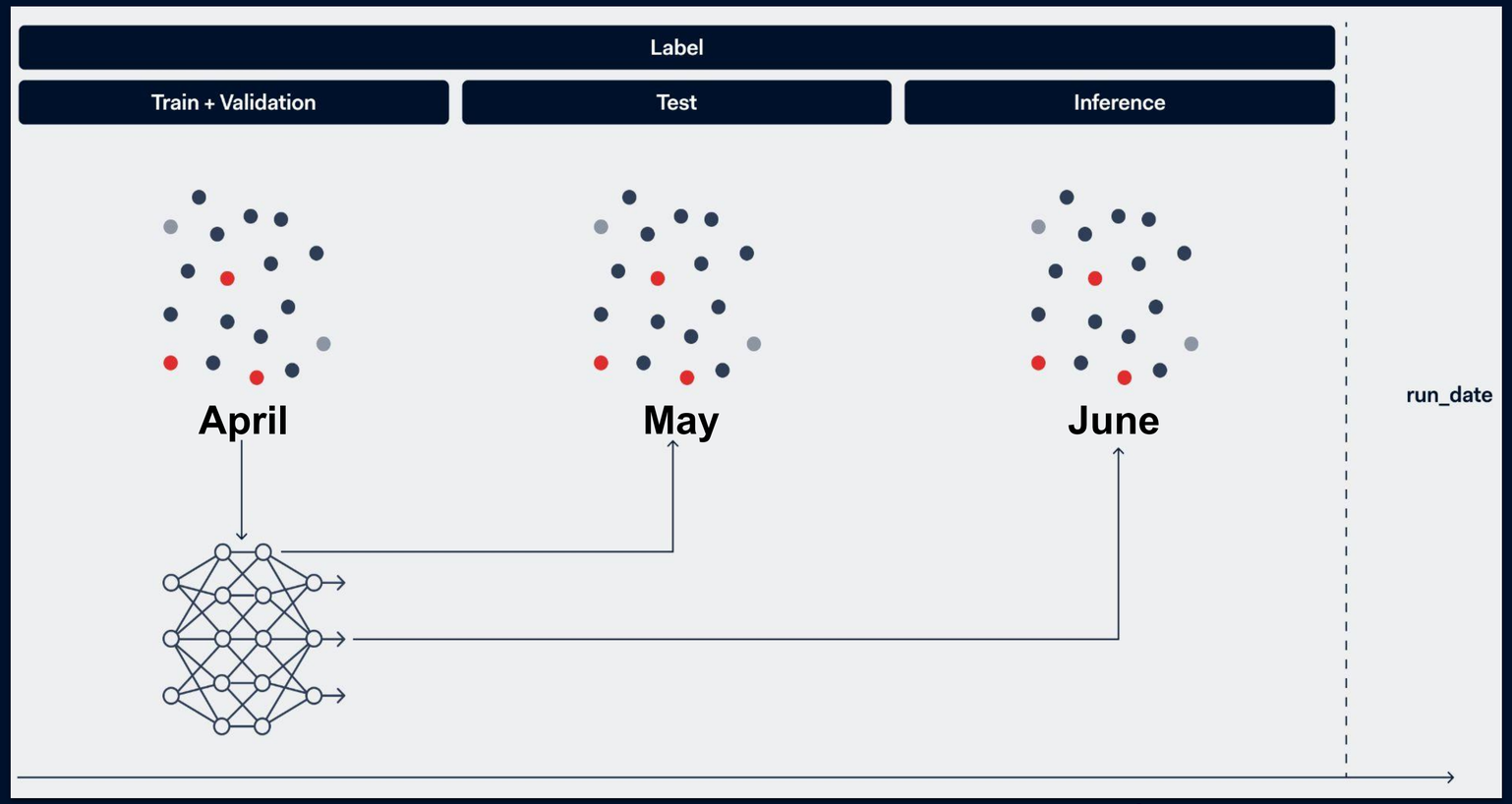
Takeaways:

1. 3 graphs were generated for training, testing and inference

adyen

engineered
for ambition

Model | Training approach



Takeaways:

1. 3 graphs were generated for training, testing and inference
2. The data is splitted by time, e.g. training graph is from April, testing graph is from May, inference graph is from June.

adyen

engineered
for ambition

GNNs in production

GNNs in production

Pipeline overview

Adyen Payment Platform | Big Data Platform | Customer Area

GNNs in production

Pipeline overview

Adyen Payment Platform

Big Data Platform

Customer Area

GNNs in production

Pipeline overview

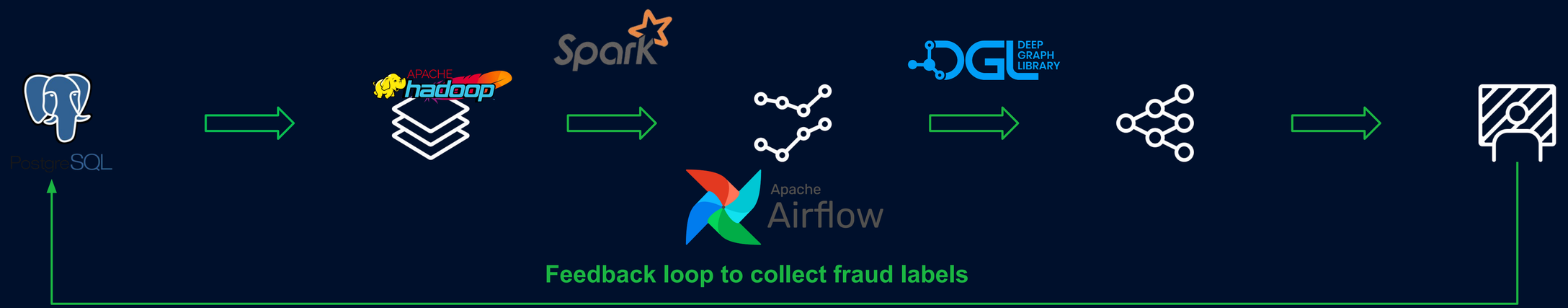
Adyen Payment Platform

Big Data Platform

Customer Area

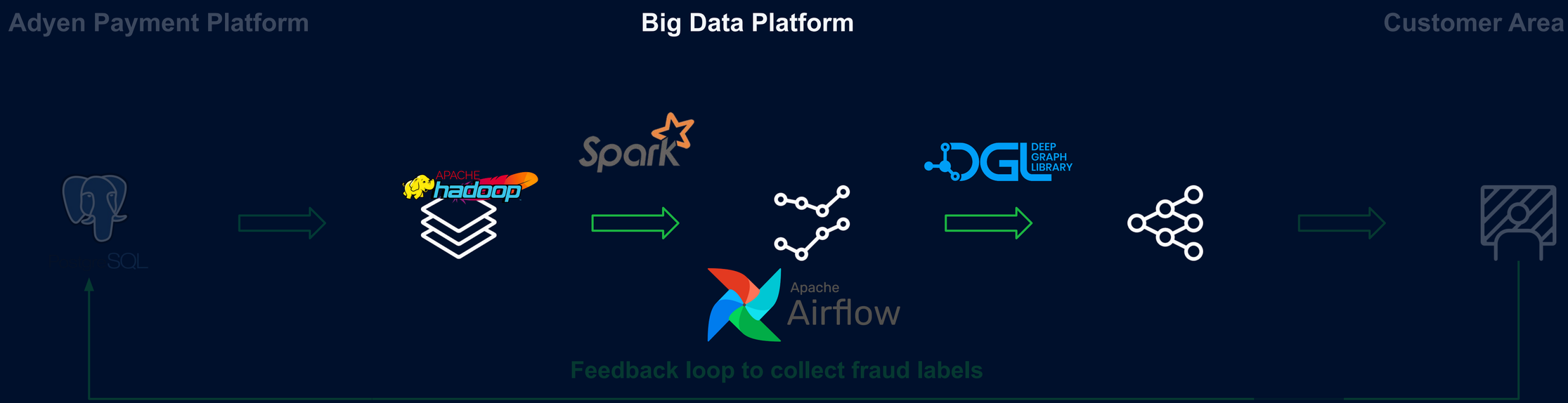
GNNs in production

Pipeline overview



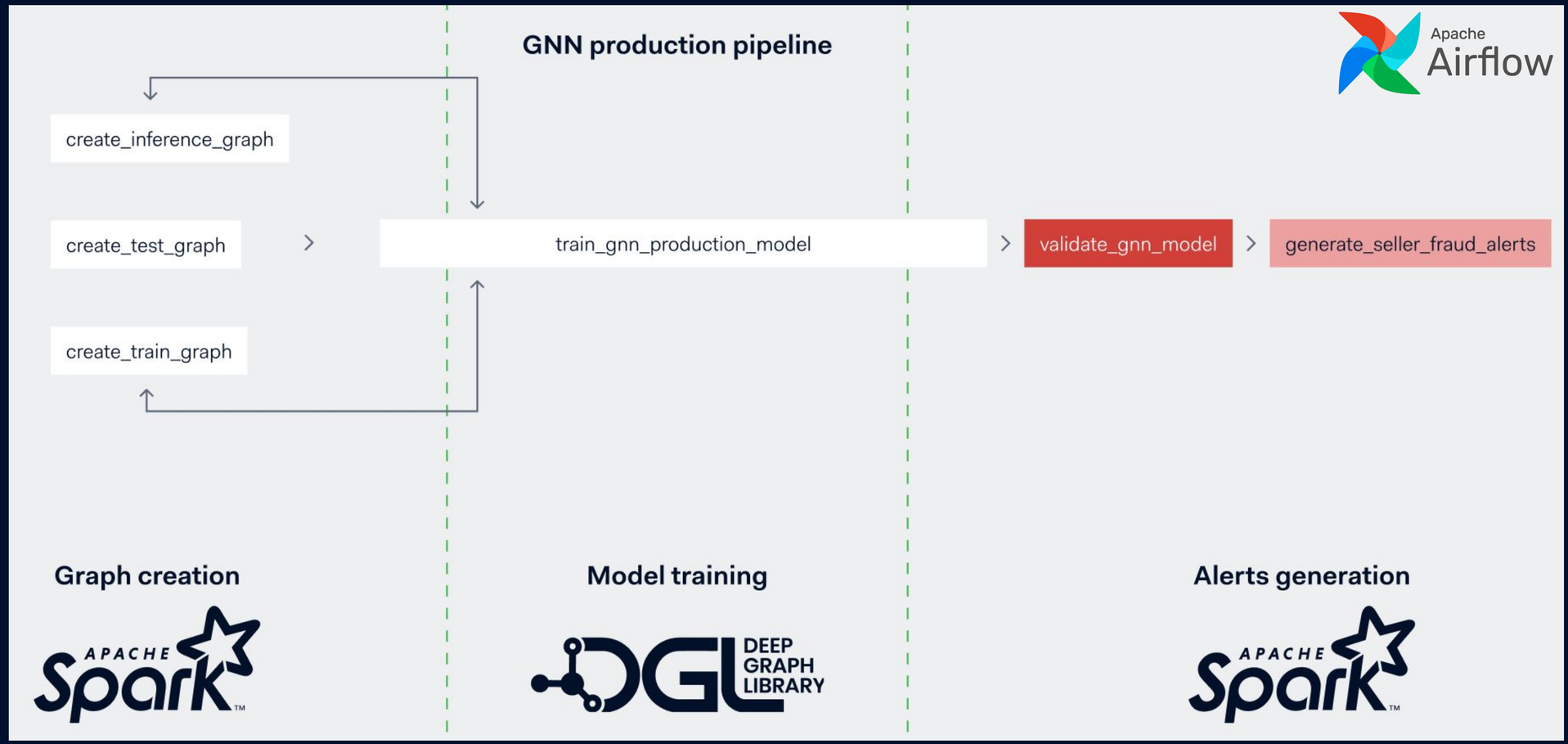
GNNs in production

Big Data part



GNNs in production

Zoom into GNN component on airflow



GNNs in production

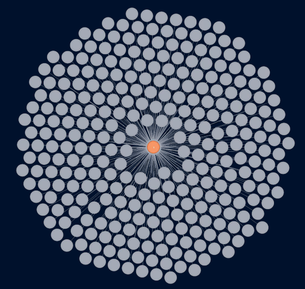
Tech choices

Aspect	Our choice	Alternatives
Graph Creation	❑ Spark GraphFrames *Ready to go for production* *Graph and feature creation in a batch mode*	❑ Graph Databases *Real time updates and graph queries*
Model Training	❑ DGL *Distributed training and graph partitioning*	❑ PyG

Lessons learned

Lessons learned

Super nodes / nodes with lots of neighbors



- **Possible reasons of being a super node:**
  - A special “seller”, e.g., merchant official accounts
  - Data integration issue, e.g., shopper@123.com
- **Issues with having super nodes in your graph**
  - Huge edge table -> not scalable
  - Significant performance drop
- **Takeaways:**
  - Graph cleaning with business logic
  - Graph validation in production

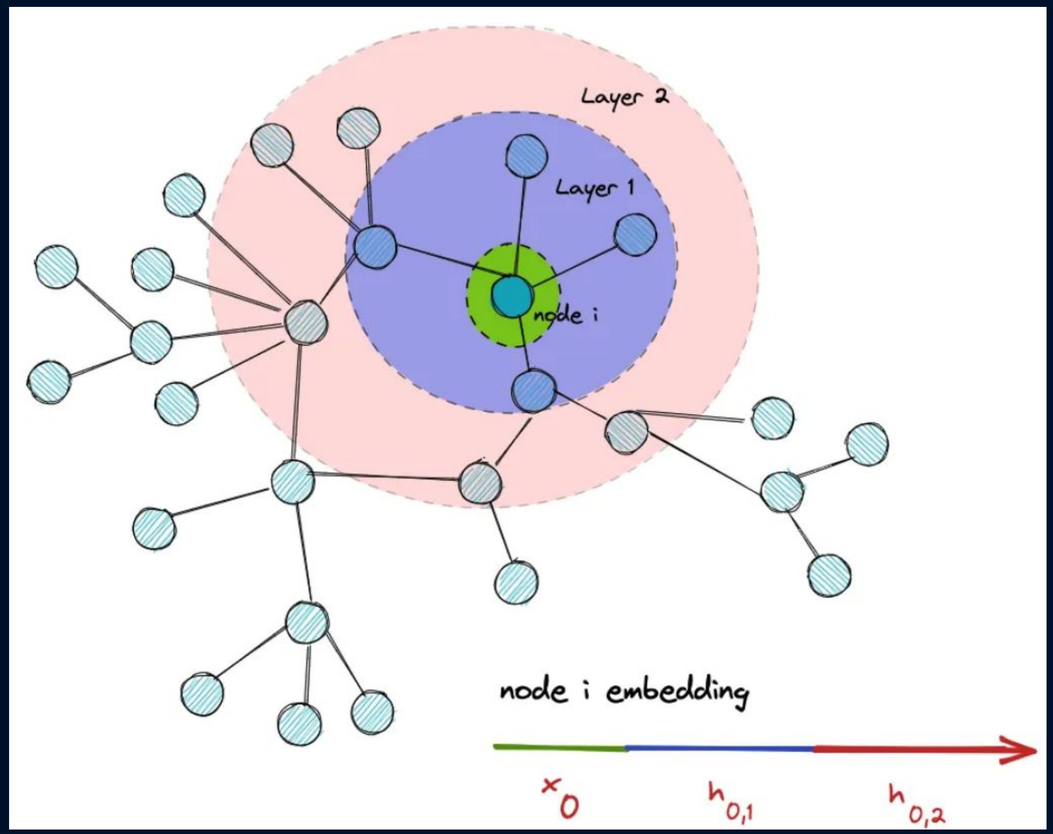
adyen

engineered
for ambition

Lessons learned

Deep or shallow?

- **Over-smoothing issue**
  - ➢ GNN treats all nodes as if they were identical
- **Takeaway: start with shallow network**



adyen

engineered
for ambition

Figure is from blog <Over-smoothing issue in graph neural network>

Lessons learned

Reproducibility matters

```
def set_determinism(deterministic: bool):
    if deterministic:
        random.seed(0)
        np.random.seed(0)
        torch.manual_seed(0)
        torch.use_deterministic_algorithms(True)
        dgl.seed(0)
```

adyen

engineered
for ambition

On a final note

Was GNN worth it?

- **Putting GNNs in production was a big investment**
  - Standardized and scalable graph API to handle tabular data
  - Steep learning curve to build up the knowledge around GNNs

adyen

engineered
for ambition

On a final note

Was GNN worth it? Yes!! 

- **Putting GNNs in production was a big investment**
  - Standardized and scalable graph API to handle tabular data
  - Steep learning curve to build up the knowledge around GNNs

- **Big business impact**
  - GNN outperformed all the existing business rules + models by a big margin
  - Safeguarded both merchants and Adyen from seller frauds

adyen

engineered
for ambition

Adyen Knowledge Hub

Q&A

Thank you!

adyen

engineered
for ambition

References

- **GCN:** Semi-Supervised Classification with Graph Convolutional Networks
- **Batch Normalization:** Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift
- **GraphSage:** Inductive Representation Learning on Large Graphs