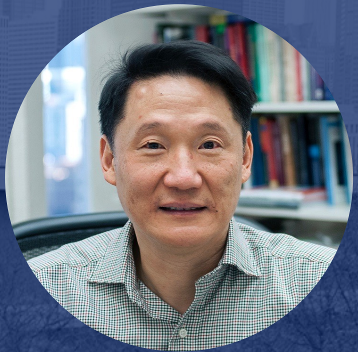
Leveraging Graph Neural Networks for Financial Fraud Detection: Practices and Challenges

Yingtong Dou

ydou5@uic.edu

Kay Liu

zliu234@uic.edu



Philip S. Yu

psyu@uic.edu

University of Illinois Chicago

Machine Learning in Finance Workshop @ KDD 2022

Tutorial Outline

- **Part 1 & 2 (Yingtong)**
  - **Background: Financial fraud detection and graph neural networks.**
  - **Supervised methods and DGFraud.**

- **Part 3 & 4 (Kay)**
  - **Unsupervised methods, PyGOD, and benchmark.**
  - **ML pipelines, and live demo.**

- **Part 5 (Yingtong)**
  - **Challenges, solutions, insights, guideline, and resources.**

Tutorial Outline

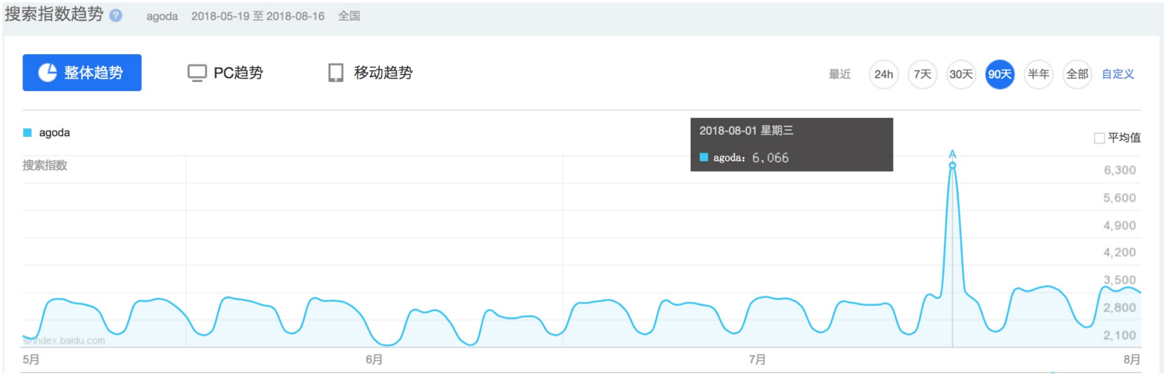
- Part 1 & 2 (Yingtong)
  - Background: Financial fraud detection and graph neural networks.
  - Supervised methods and DGFraud.
- Part 3 & 4 (Kay)
  - Unsupervised methods, PyGOD, and benchmark.
  - ML pipelines, and live demo.
- Part 5 (Yingtong)
  - Challenges, solutions, insights, guideline, and resources.

What is Fraud?

- **Fraud definition according to U.S. Law:**
  - a misrepresentation of a fact, made from one person to another, with knowledge of its falsity and for the purpose of inducing the other to act.

- **Fraudster vs. Hacker**
  - Most fraudsters are NOT hackers.
  - Only few hackers are fraudsters.

- **Fraud vs. Anomaly**
  - Not all frauds are anomalies.
  - Not all anomalies are frauds.



Machine Learning in Financial Fraud Detection

Identity-based Detectors

Demographical information

Behavior-based Detectors

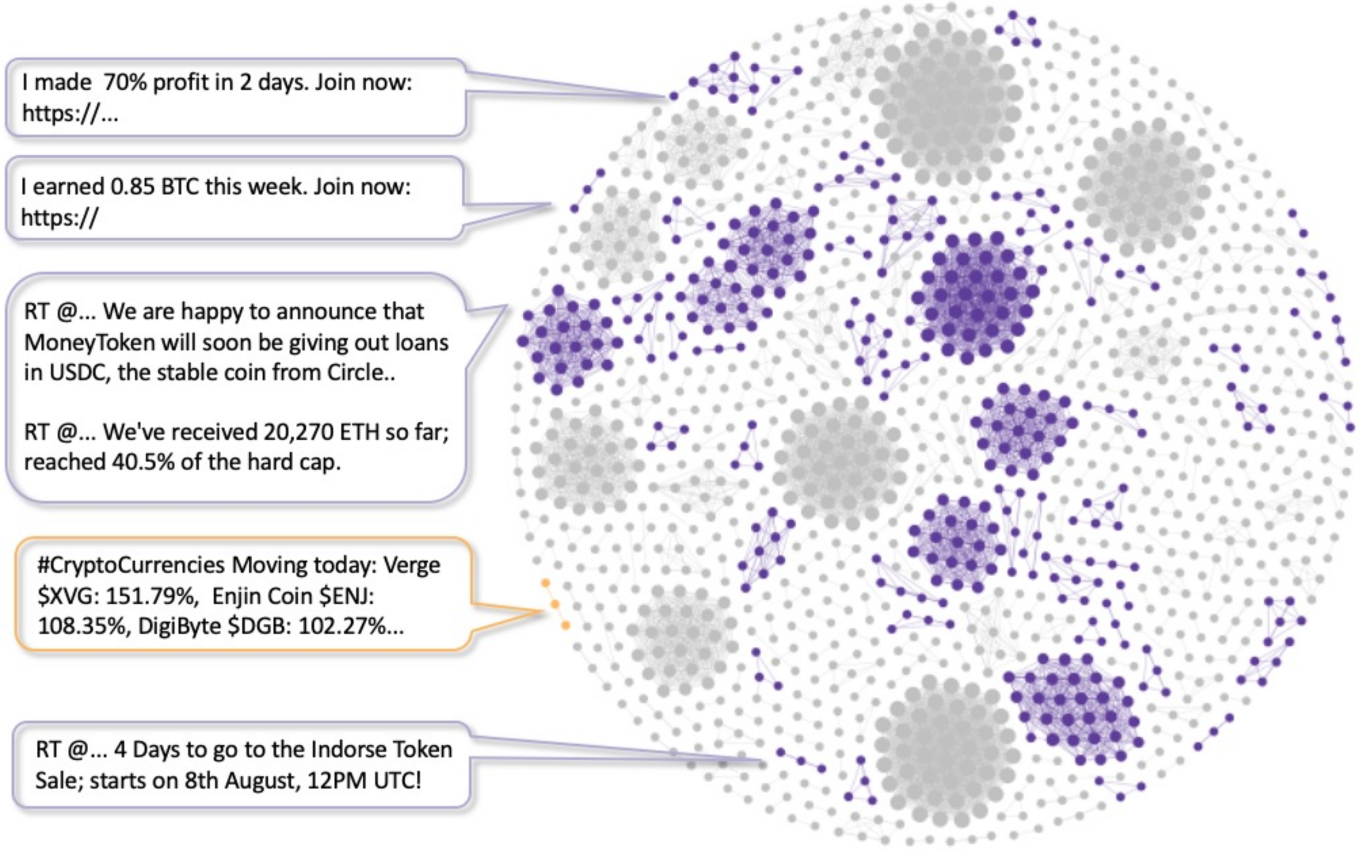
The frequency of transaction

Graph-based Detectors

Discuss in this tutorial!

Graph-based Fraud Detection

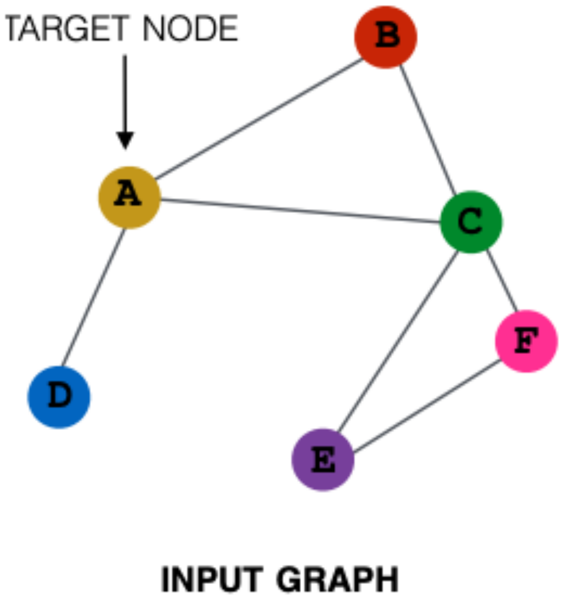
Bot Account on Twitter

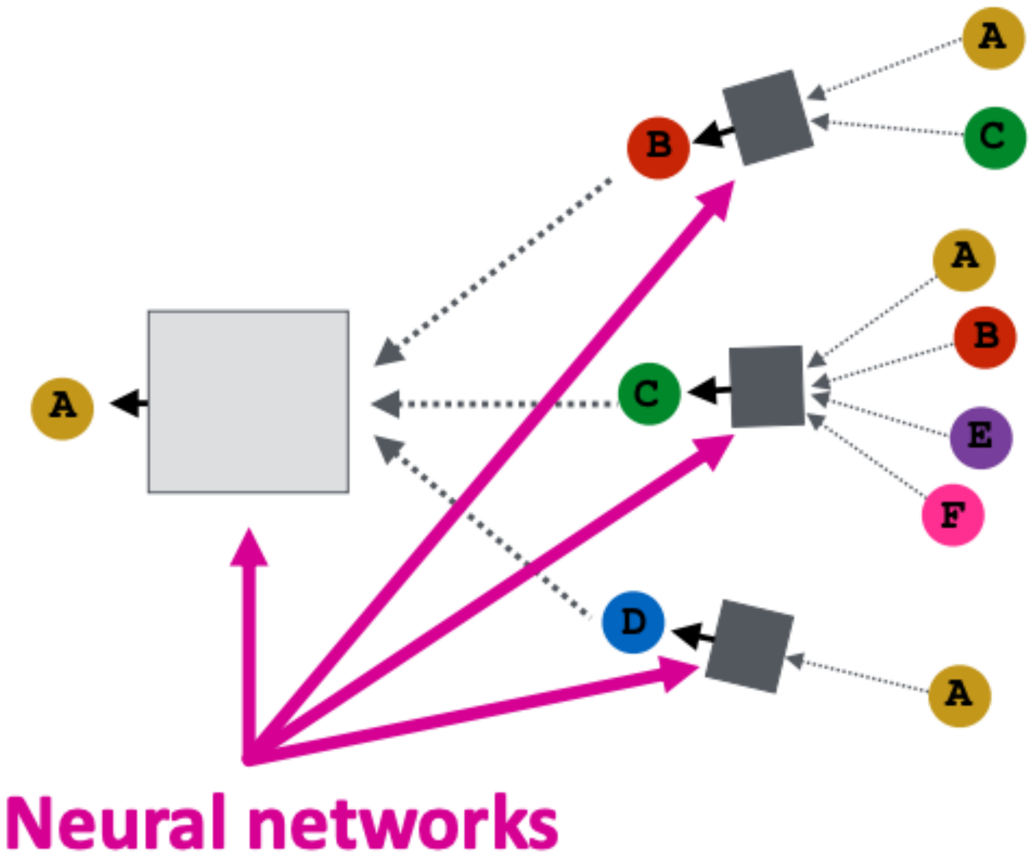


Coordinated Accounts on Social Network[1]

[1] Pacheco, Diogo, et al. "Uncovering Coordinated Networks on Social Media: Methods and Case Studies." *ICWSM*. 2021.

Graph Neural Networks





Key idea: the connected nodes are similar (homophily assumption)

Image from Jure Leskovec, Stanford CS224W: Machine Learning with Graphs, http://cs224w.stanford.edu

GNN Use Cases in Industry

- Pinterest, Snapchat
  - Recommender systems
- Amazon & United Airlines
  - Information extraction
- AstraZeneca
  - Molecular Generation

GNN for Financial Fraud

- **Insurance Fraud**
- **Loan Defaulter**
- **Money Laundering**
- **Credit Card Fraud**
- **Transaction Fraud**
- **Cash-out Fraud**
- **Bitcoin Fraud**

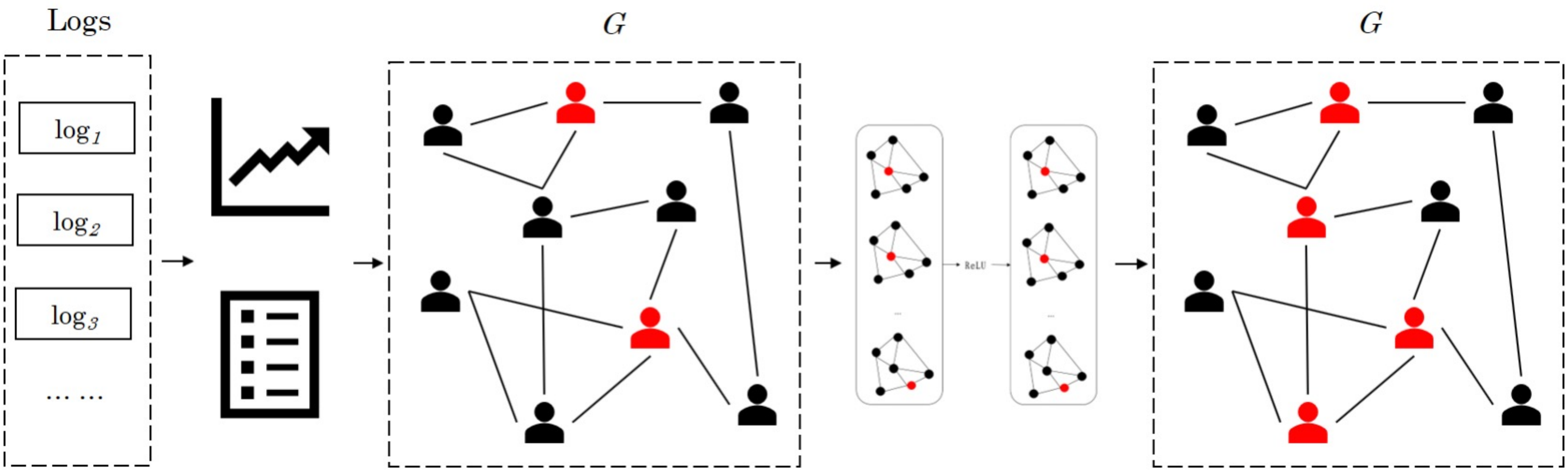
Tutorial Outline

- Part 1 & 2 (Yingtong)
  - Background: Financial fraud detection and graph neural networks.
  - Supervised methods and DGFraud.

- Part 3 & 4 (Kay)
  - Unsupervised methods, PyGOD, and benchmark.
  - ML pipelines, and live demo.

- Part 5 (Yingtong)
  - Challenges, solutions, insights, guideline, and resources.

Supervised GNN



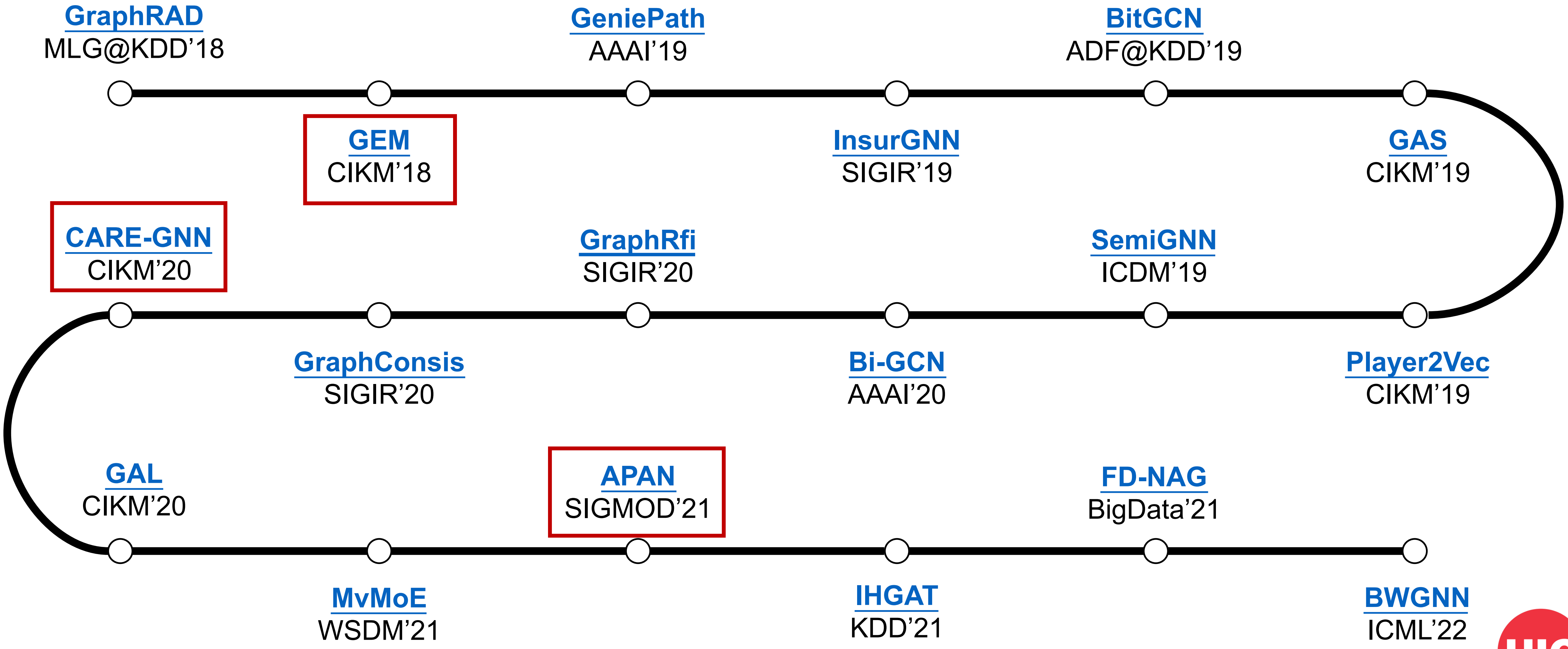
(1) Graph Construction.

(2) Training GNN on the Graph with labeled nodes.

(3) Classifying Unlabeled Nodes.

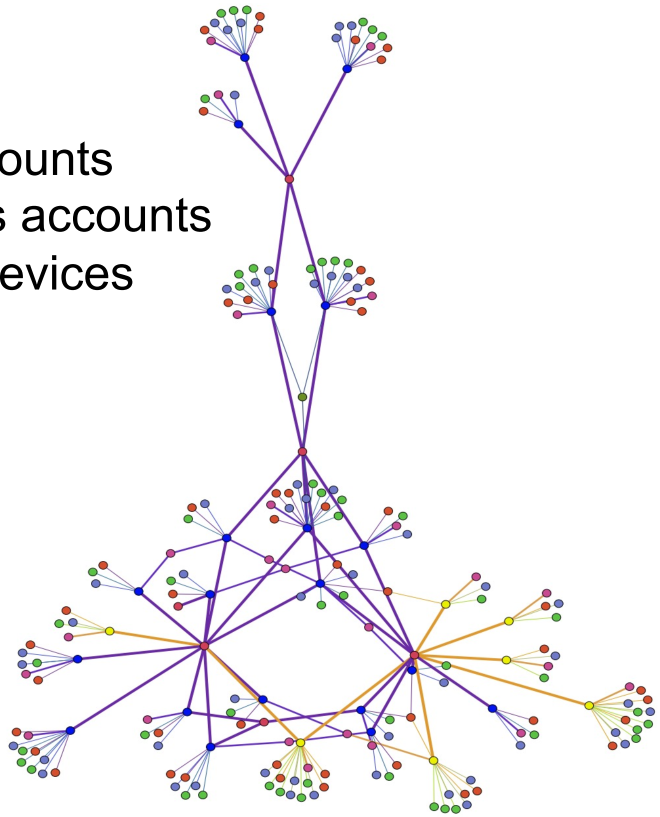
Image from Jianyu Wang et. al. 2019. FdGars: Fraudster Detection via Graph Convolutional Networks in Online App Review System. WWW 2019.

A Short History of GNN Fraud Detection



GEM (CIKM'18)

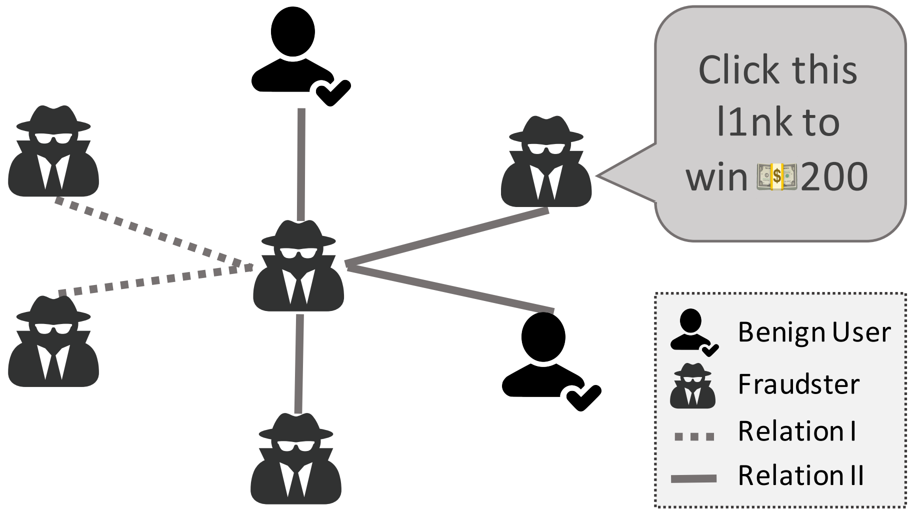
Blue: normal accounts
Yellow: malicious accounts
Other: different devices



Account-Device Heterogeneous Graph

- Task: malicious accounts detection in mobile payment service (Alipay).
- **The first paper leveraging the heterogeneous graphs for fraud detection.**
- Device types include UMID, MAC address, IMSI, APDID (Alipay Fingerprint).
- Code is available.

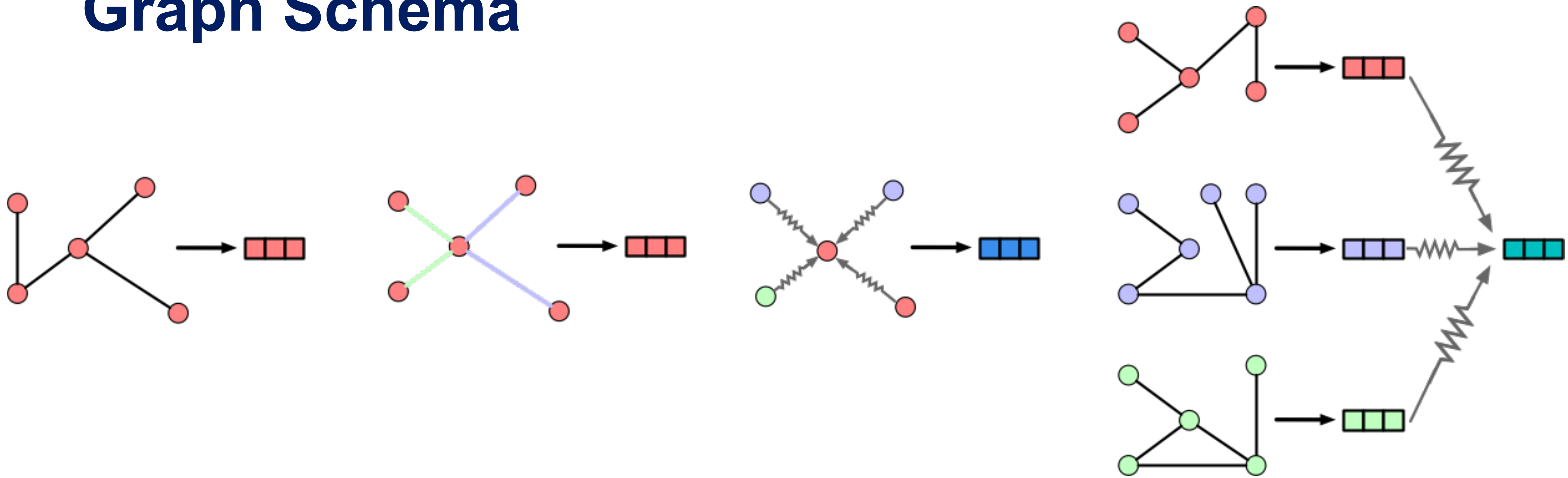
CARE-GNN (CIKM'20)



Fraudster Camouflage
&
Multi-relational Graph

- Task: spam review detection on Yelp; malicious reviewer detection on Amazon.
- **Top 10 influential papers in CIKM'20.**
- Using reinforcement learning to select the most informative neighbors for GNNs.
- Code is available.

Graph Schema



Homogeneous

BitGCN
FdGars
GeniePath
FD-NAG

Multi-relation

GraphConsis
CARE-GNN
PC-GNN

Heterogeneous

GAS
mHGNN
IHGAT

Hierarchical

GEM
SemiGNN
Player2Vec
AA-HGNN

Graph Schema is Crucial

- Task: finding fraud colluders on an online insurance platform.
- The suspicious signal can only be visible under certain graph schemas.
- Graph schema design is the key step for applied graph machine learning.

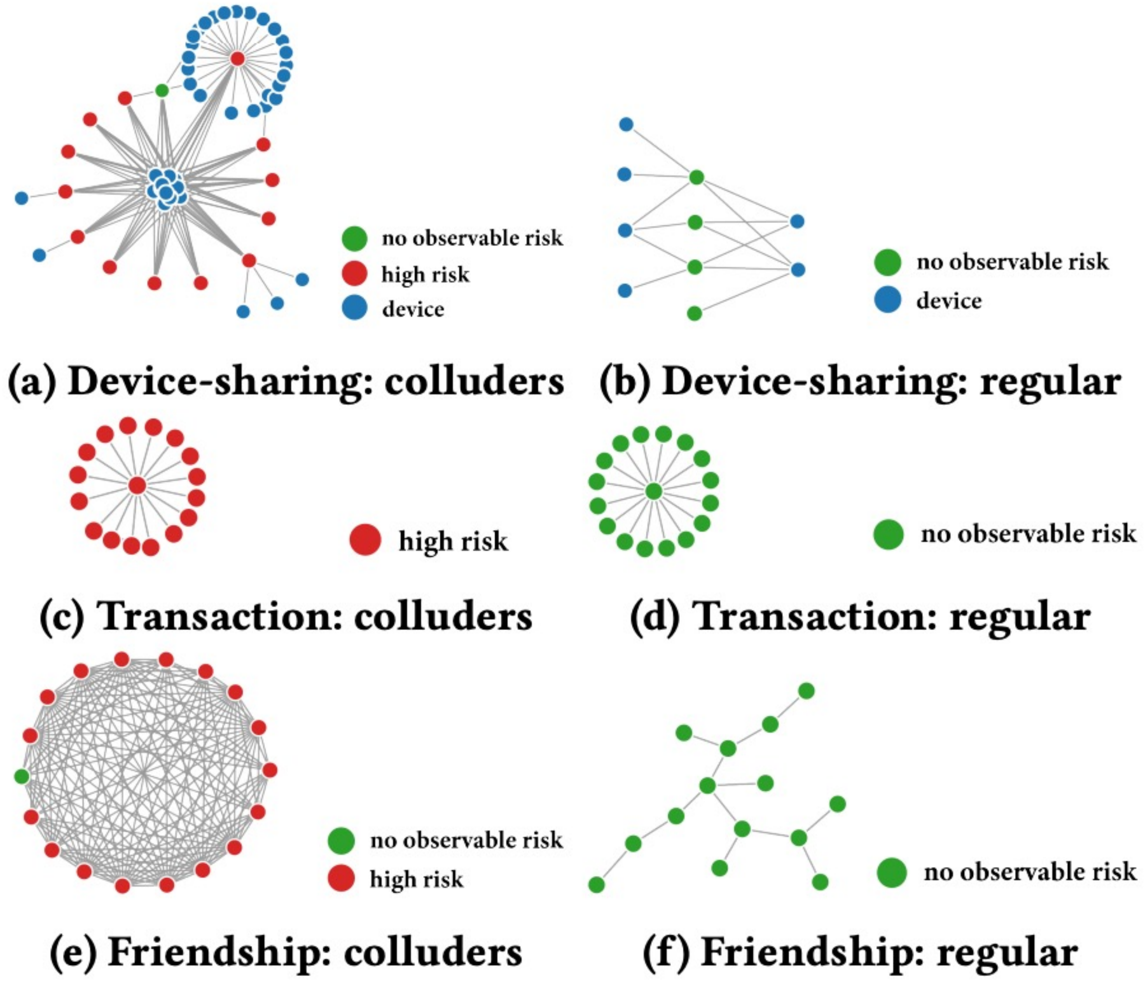


Image from Liang, Chen, et al. "Uncovering insurance fraud conspiracy with network learning." *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019.

Dynamic GNN: A Low-latency Inference Solution

- Each node has a mailbox (memory) to store the up-to-date neighbor information.
- The inference can be done by aggregating information from the memory.
- The memory can be updated asynchronously.
- APAN: e-commerce transaction fraud detection.

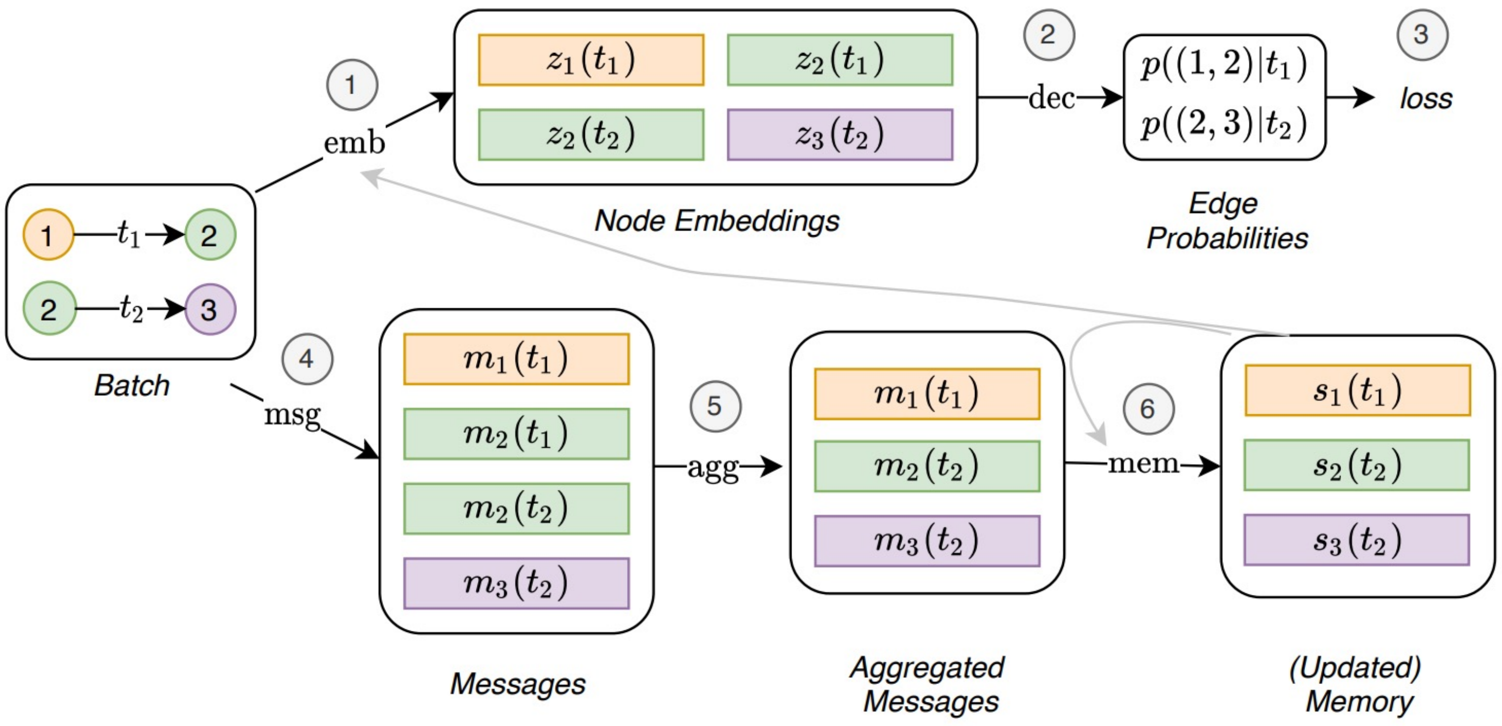


Image from Rossi, Emanuele, et al. "Temporal graph networks for deep learning on dynamic graphs." *arXiv preprint arXiv:2006.10637* (2020).

DGFraud

- DGFraud – A Deep Graph-based Toolbox for Fraud Detection.

DGFraud

build passing | license Apache-2.0 | release v0.1.0 | PRs welcome

DGFraud-TF2

build passing | tensorflow 2.X | python 3.6 | 3.7 | 3.8 | 3.9 | PRs welcome | release v0.1.0

Combined 550+ Stars on GitHub

Model	Application	Graph Type	Base Model
SemiGNN	Financial Fraud	Heterogeneous	GAT, LINE, DeepWalk
Player2Vec	Cyber Criminal	Heterogeneous	GAT, GCN
GAS	Opinion Fraud	Heterogeneous	GCN, GAT
FdGars	Opinion Fraud	Homogeneous	GCN
GeniePath	Financial Fraud	Homogeneous	GAT
GEM	Financial Fraud	Heterogeneous	GCN
GraphSAGE	Opinion Fraud	Homogeneous	GraphSAGE
GraphConsis	Opinion Fraud	Heterogeneous	GraphSAGE
HACUD	Financial Fraud	Heterogeneous	GAT

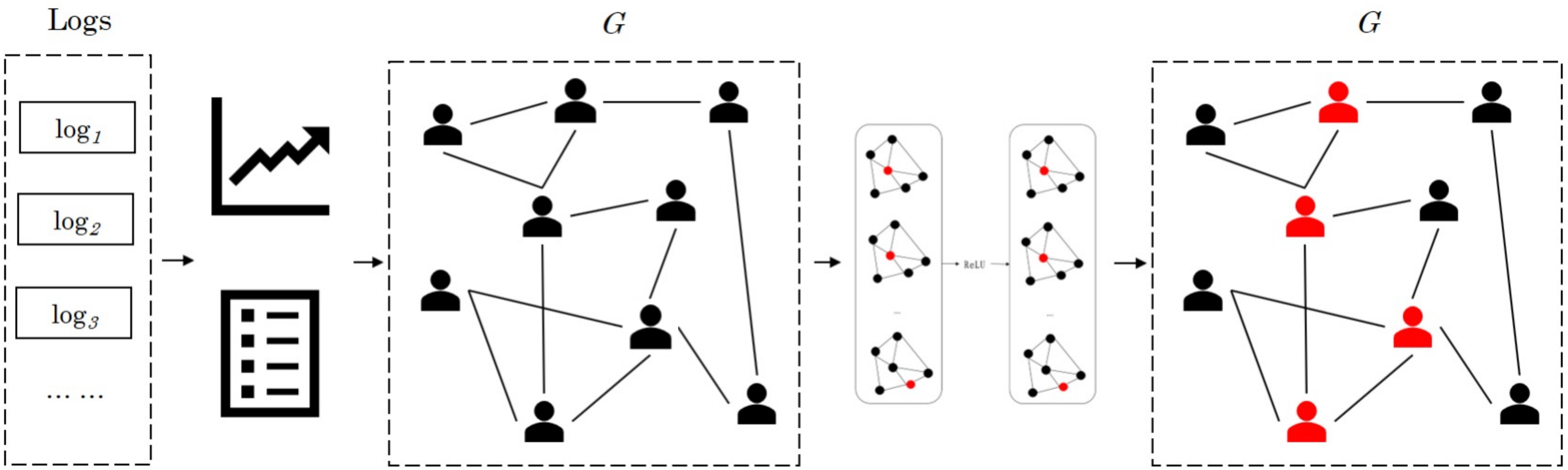
Tutorial Outline

- Part 1 & 2 (Yingtong)
  - Background: Financial fraud detection and graph neural networks.
  - Supervised methods and DGFraud.
- Part 3 & 4 (Kay)
  - Unsupervised methods, PyGOD, and benchmark.
  - ML pipelines, and live demo.
- Part 5 (Yingtong)
  - Challenges, solutions, insights, guideline, and resources.

Unsupervised Fraud Detection with Graphs

- **Label scarcity**
  - Ground truth labels can be expensive, even impossible to obtain.
- **Novelty detection**
  - Unsupervised learning does not reply on existing labeled data.
- **Preprocessing for downstream tasks**
  - E.g., Outlier resistant node classification.

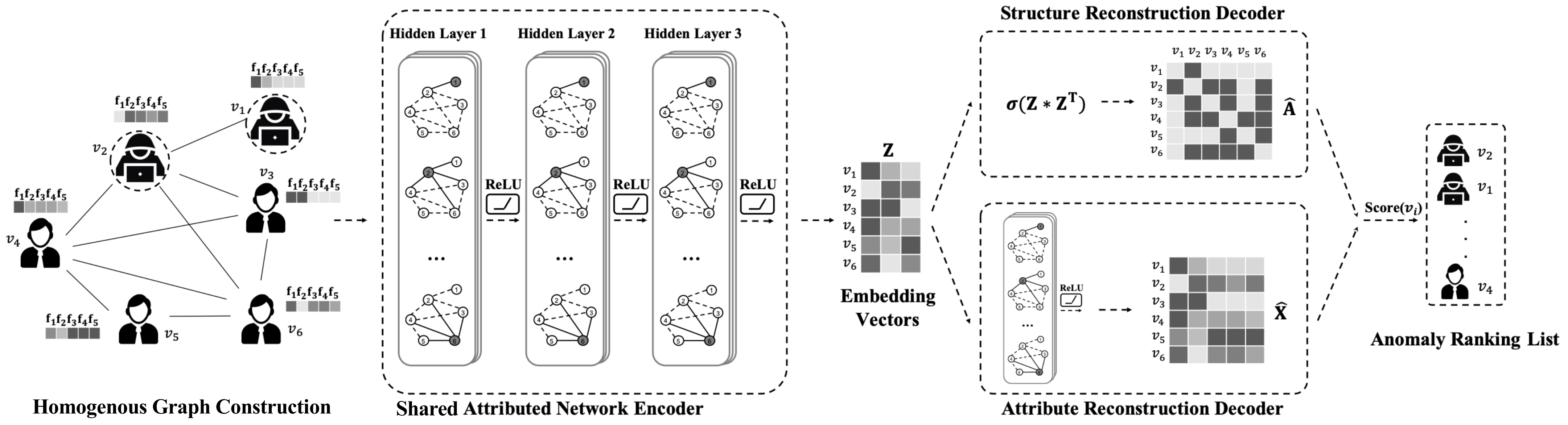
Graph Auto-Encoder (GAE)



(1) Graph Construction. (2) Training GAE on the Graph. (3) Detecting Outlier Nodes.

Modified from Jianyu Wang et. al. 2019. FdGars: Fraudster Detection via Graph Convolutional Networks in Online App Review System. WWW 2019.

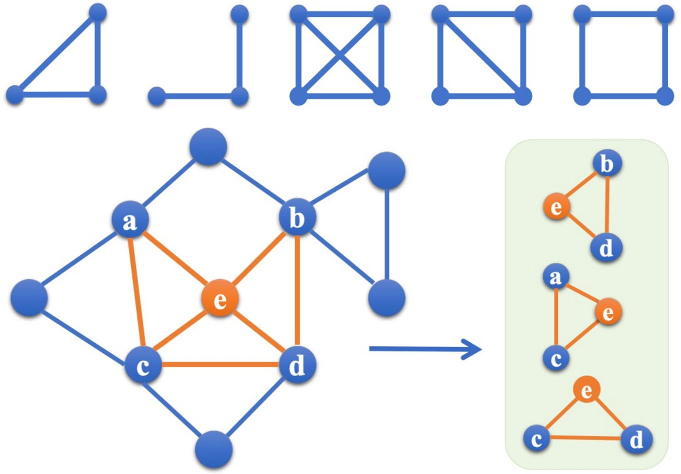
DOMINANT (SDM'19)



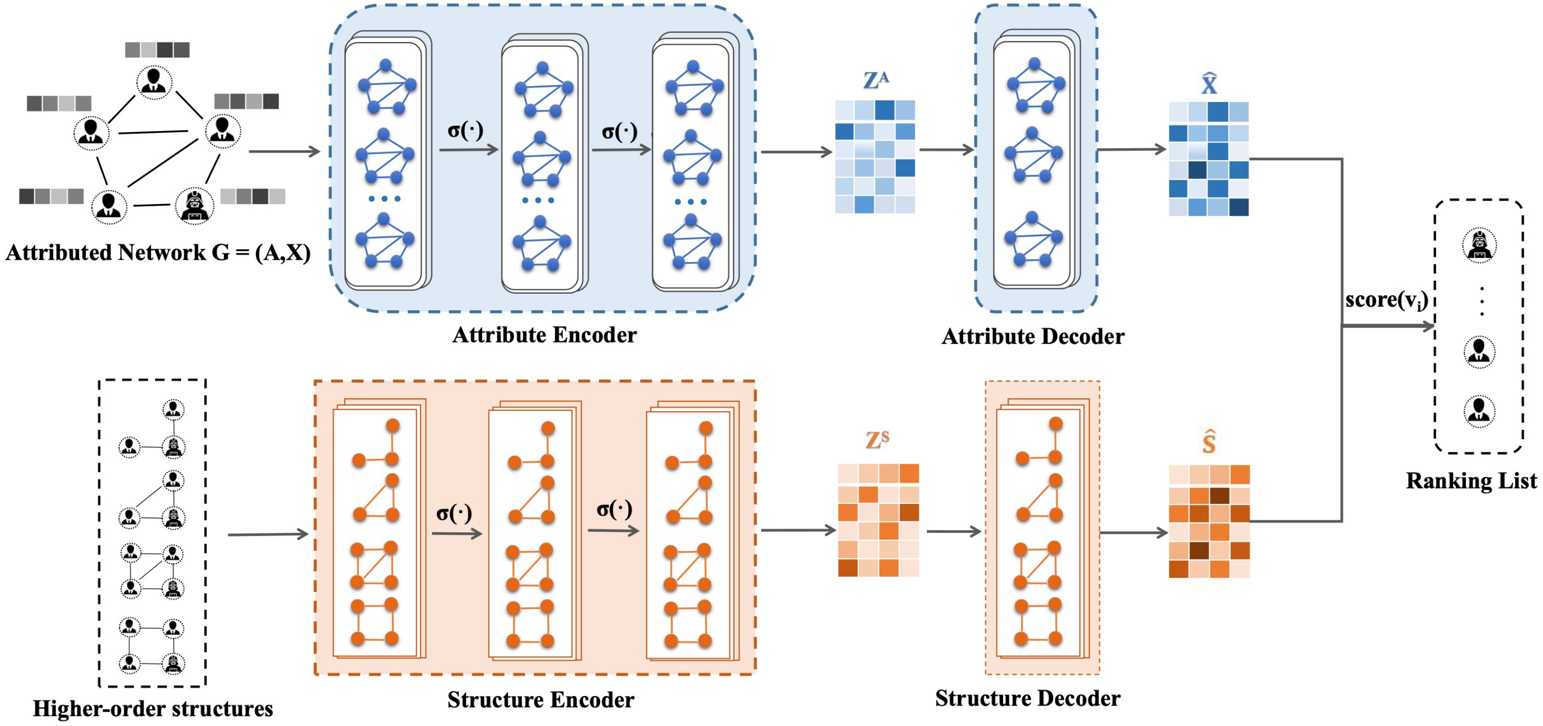
- The first attempt of graph auto-encoder in graph anomaly detection problem.
- Adopted multi-task learning framework to jointly detect anomalies from two aspects.
- Using reconstruction error of structure and attribute as anomaly score.

GUIDE (Big Data’21)

Node Motif Degree

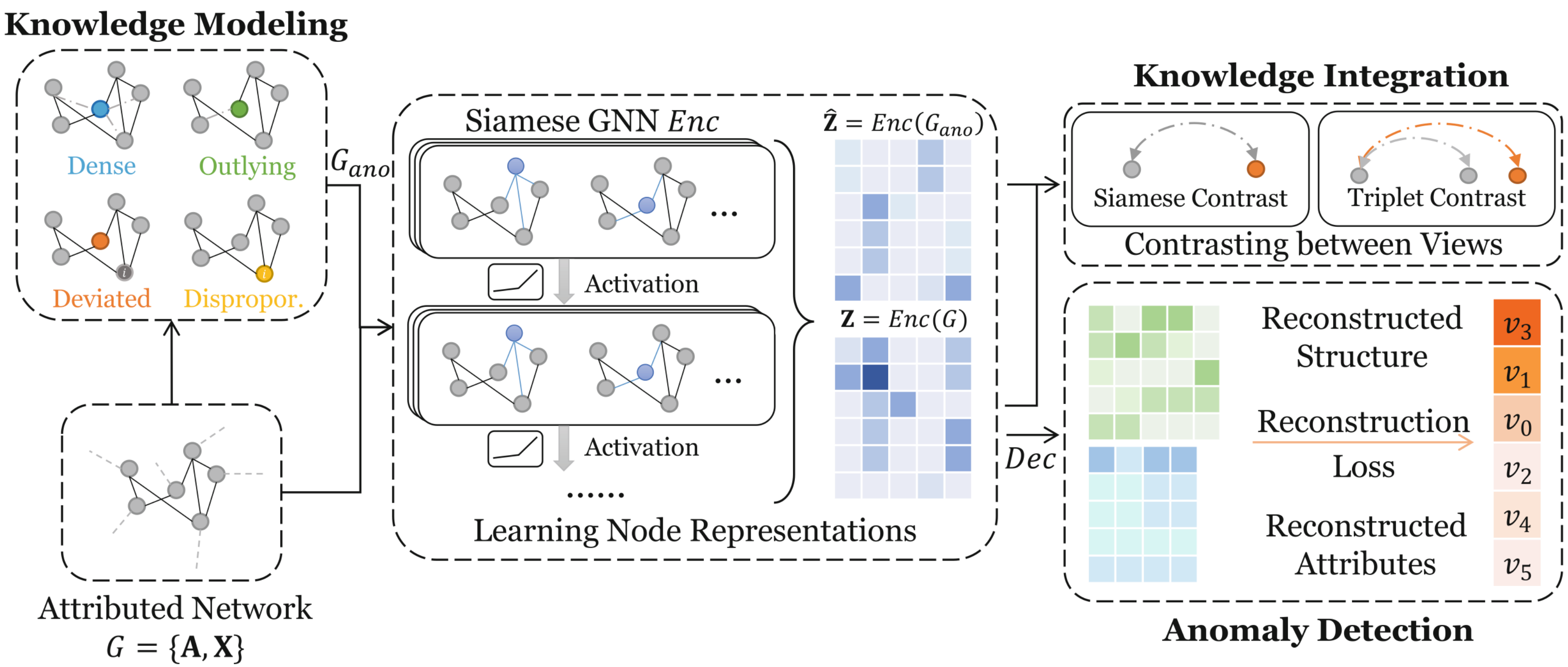


NodeMotifDegree(e,) = 3



- Capture higher-order structure information with node motif degree.
- Largely improve the scalability for AE, but huge burden on node motif degree counting.
- Imprecise estimate (e.g., LRP) can accelerate node motif degree counting.

CONAD (PAKDD'22)



- Data augmentation with predefined anomaly knowledge modeling.
- Appling contrastive learning in graph anomaly detection problem.
- Knowledge integration with Siamese Contrast and Triplet Contrast.

PyGOD

A Python Library for Graph Outlier/Anomaly Detection

Detecting graph outliers in 5 lines of code

Received 600+ Stars on GitHub

Homepage: https://pygod.org
Doc: https://docs.pygod.org
Software Paper: https://arxiv.org/abs/2204.12095
Email: dev@pygod.org

Backbone	Abbr	Year	Sampling
MLP+AE	MLPAE	2014	Yes
Clustering	SCAN	2007	No
GNN+AE	GCNAE	2016	Yes
MF	Radar	2017	No
MF	ANOMALOUS	2018	No
MF	ONE	2019	No
GNN+AE	DOMINANT	2019	Yes
MLP+AE	DONE	2020	Yes
MLP+AE	AdONE	2020	Yes
GNN+AE	AnomalyDAE	2020	Yes
GAN	GAAN	2020	Yes
GNN+AE	OCGNN	2021	Yes
GNN+AE	CoLA (beta)	2021	In progress
GNN+AE	ANEMONE (beta)	2021	In progress
GNN+AE	GUIDE	2021	Yes
GNN+AE	CONAD	2022	Yes

UNOD Benchmark

- The first comprehensive unsupervised node outlier detection benchmark.
- Provides synthetic, injected, and organic outlier detection dataset.

Data repo: https://github.com/pygod-team/data
Benchmark Paper: https://arxiv.org/abs/2206.10071
Email: benchmark@pygod.org

Dataset	Type	#Nodes	#Edges	#Feat
'weibo'	organic	8,405	407,963	400
'reddit'	organic	10,984	168,016	64
'inj_cora'	injected	2,708	11,186	1,433
'inj_amazon'	injected	13,752	515,872	767
'inj_flickr'	injected	89,250	942,316	500
'gen_time'	generated	1,000	5,746	64
'gen_100'	generated	100	618	64
'gen_500'	generated	500	2,662	64
'gen_1000'	generated	1,000	4,936	64
'gen_5000'	generated	5,000	24,938	64
'gen_10000'	generated	10,000	49,614	64

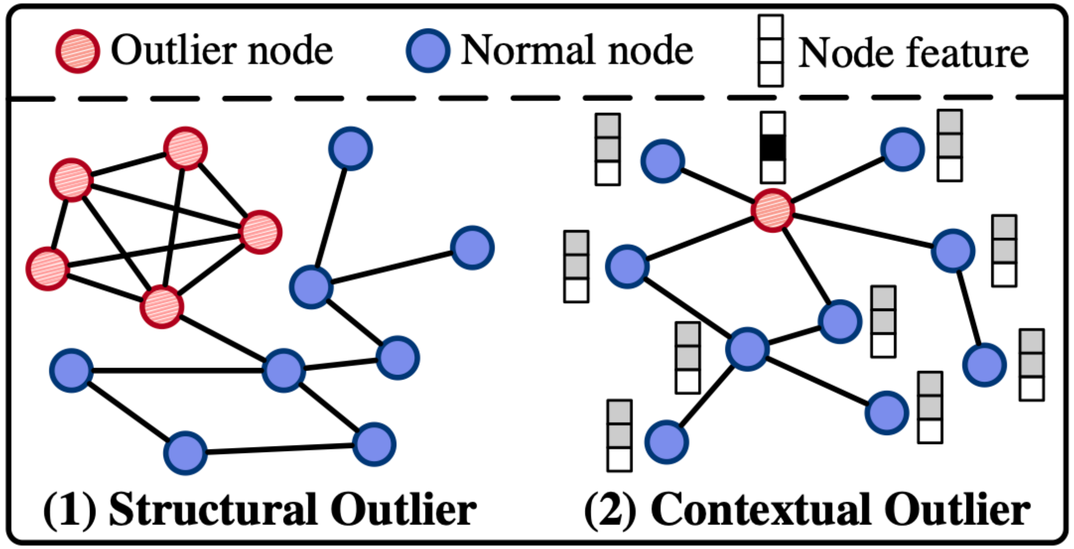
Benchmark on Performance

- Deep graph-based methods are generally better than others.
- No algorithm outperforms on all datasets in expectation.
- Performance on synthetic outliers may not generalize to organic outliers.
- Trade-off between algorithm stability and potential.

Benchmark on Outlier Types

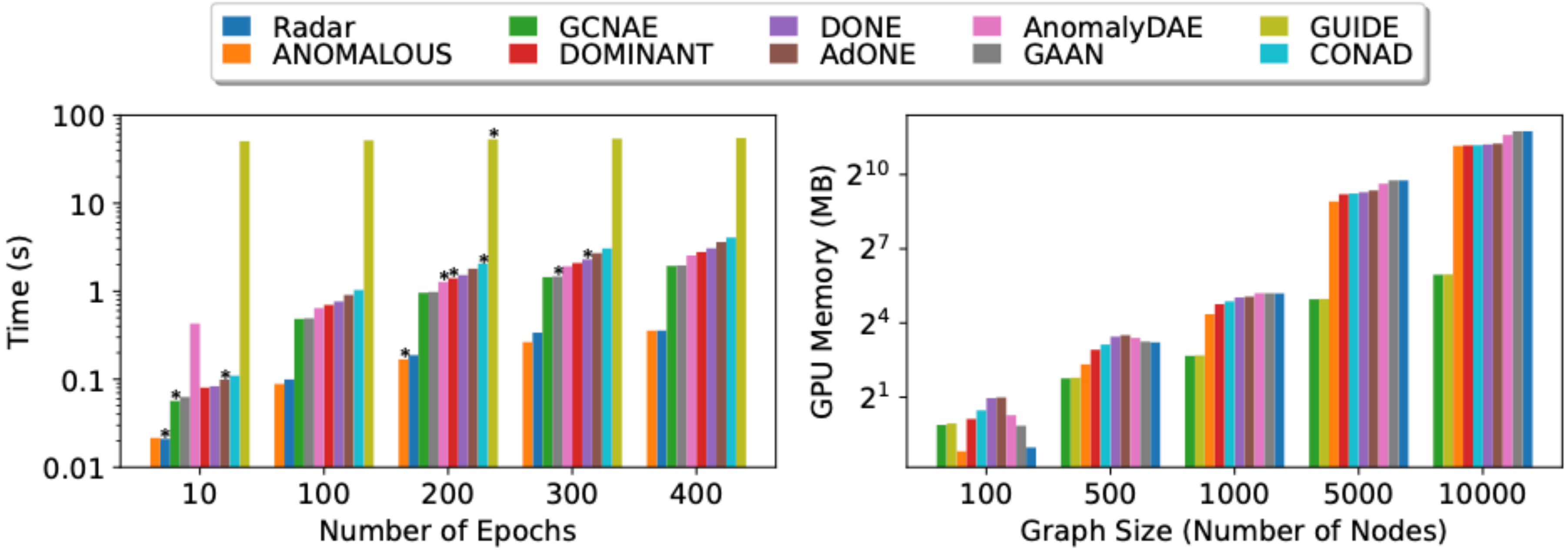
- The reconstruction instead of neighbor aggregation detects structural outlier.
- Low-order structure is sufficient for detecting structural outlier.
- None of the methods balances multiple types of outliers well.

Graph Outlier Taxonomy



Benchmark on Efficiency and Scalability

- Conventional methods are more efficient than deep methods.
- GUIDE improves scalability at an expense of efficiency.



Tutorial Outline

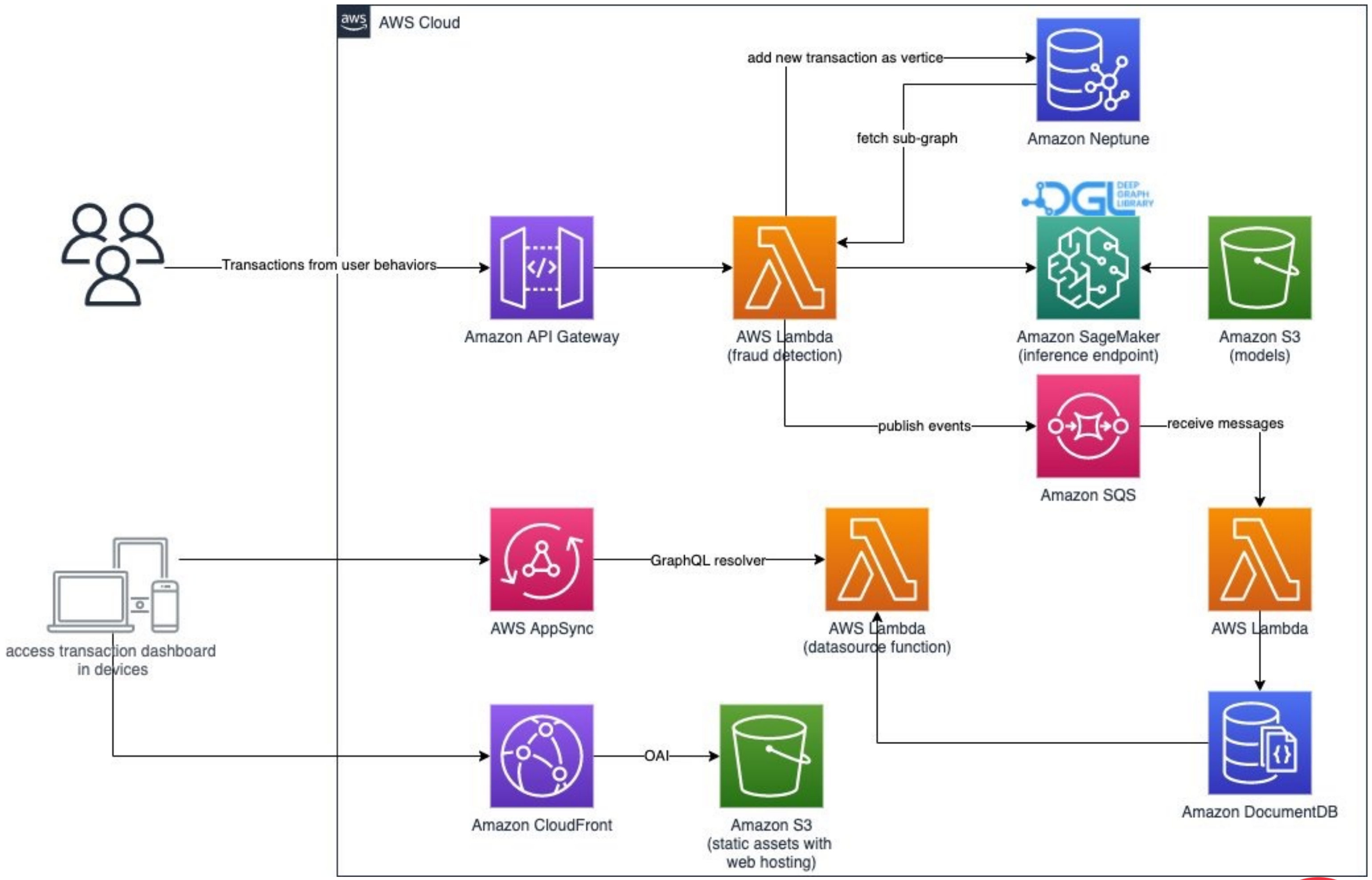
- Part 1 & 2 (Yingtong)
  - Background: Financial fraud detection and graph neural networks.
  - Supervised methods and DGFraud.

- Part 3 & 4 (Kay)
  - Unsupervised methods, PyGOD, and benchmark.
  - ML pipelines, and live demo.

- Part 5 (Yingtong)
  - Challenges, solutions, insights, guideline, and resources.

AWS DGL Real-time Fraud Detection Solution

- Processing a tabular transaction dataset into a heterogeneous graph dataset.
- Training a GNN model using SageMaker.
- Deploying the trained GNN models as a SageMaker endpoint.
- Real-time inference for incoming transactions.



TigerGraph Machine Learning Workbench

Development Framework for Graph Machine Learning

Enables data scientists to create graph neural network models and graph-enhanced models with production scale data.

Python-level Functions and Capabilities

Prepackaged Python libraries for graph data processing, graph feature engineering, subgraph sampling, data loading, and caching for out-of-DB training. No GSQL experience is required.

Compatible with Popular Machine Learning Frameworks

Work with the most popular machine learning frameworks in the market including PyTorch Geometric and DGL.

Plug-and-Play Ready Machine Learning

Flexible integration paths to works with your existing machine learning infrastructure on Amazon SageMaker, Google Vertex AI, and Microsoft Azure Machine Learning.

Tutorial Outline

- Part 1 & 2 (Yingtong)
  - Background: Financial fraud detection and graph neural networks.
  - Supervised methods and DGFraud.

- Part 3 & 4 (Kay)
  - Unsupervised methods, PyGOD, and benchmark.
  - ML pipelines, and live demo.

- Part 5 (Yingtong)
  - Challenges, solutions, insights, guideline, and resources.

GNN Financial Fraud Detection Papers

- **Online/Mobile Payment Fraud**
  - CIKM’18, ICDM’19, AAAI’19 (cash-out).
- **Insurance Fraud**
  - SIGIR’19.
- **Blockchain/Crypto Fraud**
  - MLF@KDD’19, WWW’22.
- **Loan Defaulting, Loan Fraud**
  - CIKM’20(1), CIKM’20(2), AAAI’21, WSDM’21, WWW’21, SDM’21, arXiv’22.
- **Transaction Fraud (e-commerce and credit card)**
  - arXiv’20, SIGMOD’21, KDD’21, WISE’21, TOIS’21, ESA’22, KDD’22.

Key Challenges and Solutions

- **Camouflage**
  - Neighboring filtering: SIGIR'20, CIKM'20, WWW'21.
  - Aware of adversarial behavior: IJCAI'20, WWW'20.
  - Active generative learning: ACL'20.
  - Bayesian edge weight inference: ACL'21.
- **Scalability**
  - GNN scalability: MLF@KDD'20.
  - Shallow graph models are more scalable: MLG@KDD'18, WWW'20.
- **Class imbalance**
  - Under-sampling: CIKM'20.
  - Neighbor selection: WWW'21.
  - Data augmentation: CIKM'20.

Key Challenges and Solutions (Cont'd)

- **Label scarcity**
  - Active learning: ICDM'20, TNNLS'21.
  - Ensemble learning: CIKM'20.
  - Meta learning: WSDM'21.
- **Label fidelity**
  - Active learning: TNNLS'21.
  - Human-in-the-loop: AAAI'20.
- **Data scarcity**
  - Data augmentation: CIKM'20, CIKM'21, ACL'20.

Novel Practices

- **Graph Pretraining (Contrastive Learning)**
  - Fraudster is distinguishable from its structural pattern.
  - TNNLS'21, SIGIR'21, arXiv'21(1), arXiv'21(2).
- **Dynamic/Temporal/Streaming Graph**
  - Historical information is useful for identifying fraudsters.
  - Efficiency and cost are bottlenecks.
  - CIKM'21, KDD'21(1), KDD'21(2), SIGMOD'21.
  - arXiv'21, SDM'21, ICDM'20, KDD'20.
  - ROLAND (KDD'22), arXiv'22.

Novel Practices (Cont'd)

- **Multi-task Learning**
  - Credit limit forecasting and credit risk predicting: WSDM'21.
  - Fraud detection and recommender system: SIGIR'20.
- **Explainable Fraud Detection**
  - Explainable fraud transaction detection: arXiv'20, KDD'21.
  - Explainable fake news detection: ACL'20.
- **Table2Graph**
  - Transforming credit card fraud tabular data to graph data: IJCAI'22.

Insights and Discussions

- Early detection, prevention vs. detection.
- Integrating knowledge graph (external knowledge).
- The longtail distribution of the fraud types.
- The concept drift and continual learning.
- The gap between academia and industry.

Applying GNNs in Fraud Detection: A Guideline

Using Graph? → Using GNN? → Which Task? → Schema Design → Which GNN?

- **Using Graph?**
  - The fraudsters share common properties.
  - The fraudsters have clustering behavior.
  - The trade off between cost and effectiveness.
- **Using GNN?**
  - The infrastructure.
  - The feature availability and feature types.
  - Integrating with other modules and tasks.
- **Which Task?**
  - Supervised:
    - Node/edge/graph/subgraph classification.
  - Unsupervised:
    - Community detection; anomaly detection.
- **Schema Design**
  - Node/edge type and node/edge feature.
  - Graph schema, node sampling.
  - Graph structure is flexible: SIGIR'19, ICDM'20.
- **Which GNN?**
  - GNN is chosen based on task and schema.
  - Simple GNN model is enough.
  - Dynamic GNNs need more efforts.
  - GAT and Graph-SAGE are commonly used.

Resources

- **Paper List**
  - Graph-based Fraud Detection Papers and Resources.
- **Datasets**
  - DGraph, Bitcoin-Fraud, IEEE-CIS, ETH-Phishing, Fraud-Benchmark.
- **Libraries**
  - PyG Temporal, PyOD, PyODDS, TODS, UGFraud.
- **Recent Surveys**
  - A Comprehensive Survey on Graph Anomaly Detection with Deep Learning.
  - Anomaly Mining - Past, Present and Future.
  - Graph Computing for Financial Crime and Fraud Detection: Trends, Challenges and Outlook.

Thanks!

Q & A

Yingtong Dou Kay Liu Philip Yu

University of Illinois Chicago

Slides PDF

Slides URL: https://ytongdou.com/files/GNN_Fin_Fraud.pdf