Robust Fraud Detection against Adversarial Fraudsters

Yingtong Dou

University of Illinois at Chicago

Email: ydou5@uic.edu
Twitter: @dozee_sim
Homepage: http://ytongdou.com
Project Page: https://github.com/safe-graph

Outline

- **Background :** fraud type and fraud detectors
- **KDD20:** spammer adversarial behavior and spamming practical effect
- **SIGIR20&CIKM20:** how to apply GNN to fraud detection problems
- **Resources:** dataset, toolbox, paper, survey, etc.
- **Discussion and Q&A**

A History of Spam

- 1990-2000: spam email, link farm
- 2000-2010: fake review, social bots
- 2010-2020: fake news, Deepfake

Handcrafted & Human

Social Network

Automatic & Machine Learning

Finance Technology

Ferrara, Emilio. "The history of digital spam." *Communications of the ACM* 62, no. 8 (2019): 82-91.

What is Fraud?

- **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

- **Fraud detection is an interdisciplinary problem**

Data Mining & Security & Machine Learning

Fraud Types in 2021

Social Network

- Spam Reviews
- Social Bots
- Misinformation
- Disinformation
- Fake Accounts
- Social Sybils
- Link Advertising

Finance

- Insurance Fraud
- Loan Defaulter
- Money Laundering
- Malicious Account
- Transaction Fraud
- Cash-out User
- Credit Card Fraud

Others

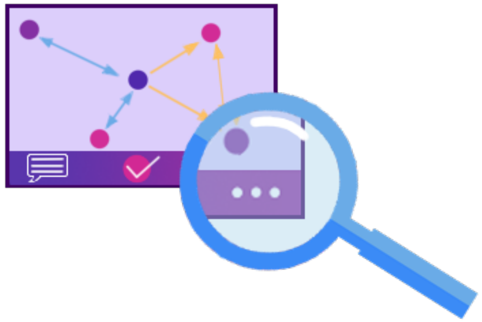
- Advertisement
- Mobile Apps
- Ecommerce
- Crowdturfing
- Promotion Abuse
- Game
- Email, Phone, SMS

Fraud Detector Types

- **Modality View:**

Content-based Detectors

Behavior-based Detectors



Graph-based Detectors

- **Technical View I:**

Rule-based Detectors **Feature-based Detectors** **Deep learning-based Detectors**

- **Technical View II:**

Unsupervised Detectors **Semi-supervised Detectors** **Supervised Detectors**

Fraudster Adversarial Behavior Example

- Elite fraudsters in Dianping[1]
  - Elite fraudsters are well organized and provide convincible reviews
- Crowd workers in Google Play[2]
  - Fraudsters will post moderate ratings to alleviate its suspiciousness
- Adversary in Tencent YingYongBao[3] and Alibaba Xianyu[4]
  - Fraudsters post reviews with symbols to evade detection
- Download fraud in Huawei App Market[5]
  - Fraud agencies can smooth their downloading frequency
- Business competitors in Amazon[6] and Yelp[7]

[1] Zheng, Haizhong, et al. "Smoke screener or straight shooter: Detecting elite sybil attacks in user-review social networks." arXiv preprint arXiv:1709.06916 (2017).
[2] Rahman, Mizanur, et al. "The Art and Craft of Fraudulent App Promotion in Google Play." Proceedings of the 2019 ACM CCS. 2019.
[3] Wen, Rui, et al. "ASA: Adversary Situation Awareness via Heterogeneous Graph Convolutional Networks." Web Conference 2020.
[4] Li, Ao, et al. "Spam review detection with graph convolutional networks." CIKM. 2019.
[5] Dou, Yingtong, et al. "Uncovering download fraud activities in mobile app markets." 2019 IEEE/ACM ASONAM, 2019.
[6] Dzieza, Josh. "Prime and punishment: Dirty dealing in the $175 billion Amazon Marketplace", The Verge, 2018.
[7] Luca, Michael, and Georgios Zervas. "Fake it till you make it: Reputation, competition, and Yelp review fraud." Management Science 62.12 (2016): 3412-3427.

KDD'20: Adversarial Behavior Modeling

Robust Spammer Detection by Nash Reinforcement Learning

Yingtong Dou
Univ. of Illinois at Chicago
ydou5@uic.edu

Guixiang Ma*
Intel Labs
guixiang.ma@intel.com

Philip S. Yu
Univ. of Illinois at Chicago
psyu@uic.edu

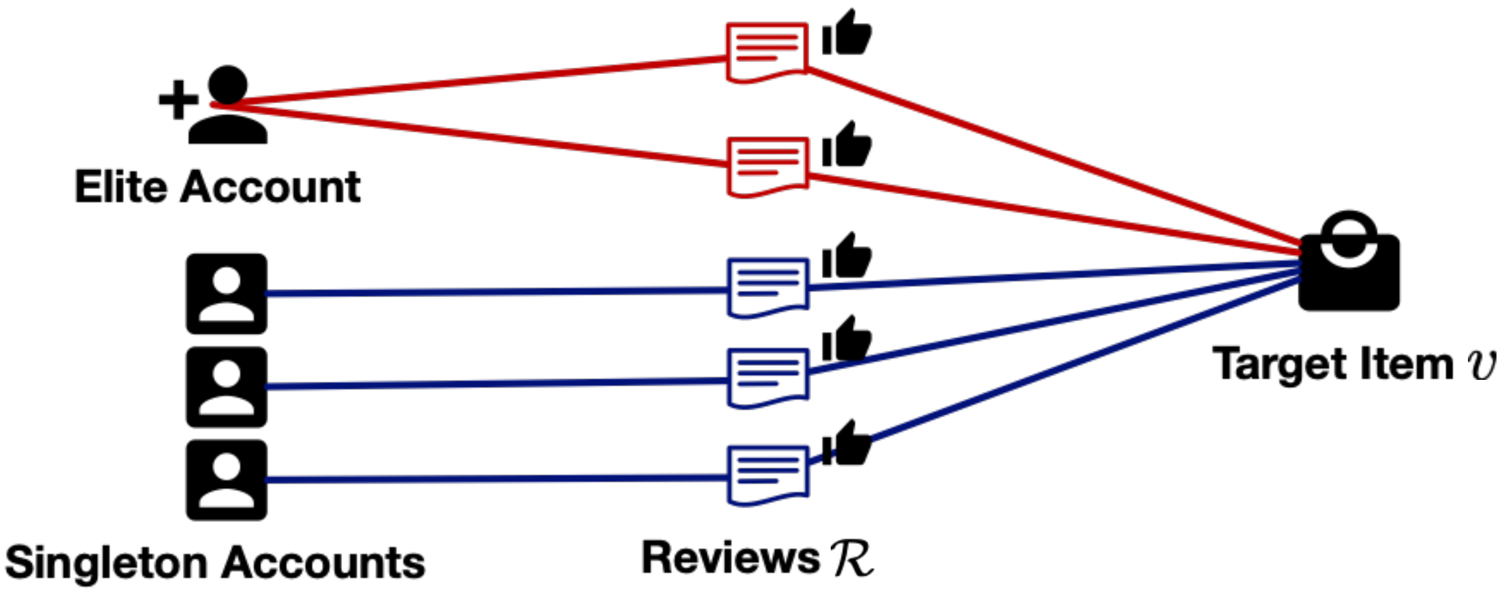
Sihong Xie
Lehigh University
xiesihong1@gmail.com

Paper: https://arxiv.org/abs/2006.06069
Code: https://github.com/YingtongDou/Nash-Detect

Turning Reviews into Business Revenues

- In Yelp, product's rating is correlated to its revenue[1]

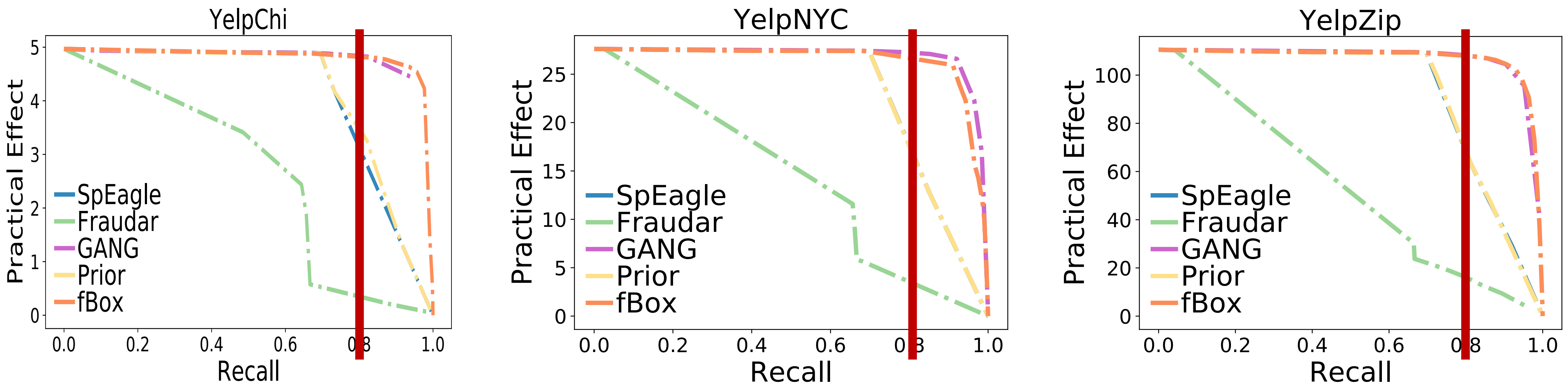
Revenue Estimation & Practical Effect : $f(v;\mathcal{R}) = \beta_0 \times \boxed{\mathrm{RI}(v;\mathcal{R})} + \beta_1 \times \boxed{\mathrm{ERI}(v;\mathcal{R}_E(v))} + \alpha$



[1] M. Luca. 2016. Reviews, reputation, and revenue: The case of Yelp. com. HBS Working Paper (2016).

Practical Effect is Better than Recall

- We run five detectors individually against five attacks
- When detector recalls are **high (>0.7)**, the practical effects are **not reduced**



Spammer's Practical Goal

Spamming Practical Effect: $\mathrm{PE}(v;\mathcal{R},\mathsf{p},\mathsf{q}) = f(v;\mathcal{R}(\mathsf{p},\mathsf{q})) - f(v;\mathcal{R})$

$f(v;\mathcal{R}(\mathsf{p},\mathsf{q}))$: Revenue after attacks

$f(v;\mathcal{R})$: Revenue before attacks

- To promote a product, the practical goal of the spammer is to **maximize** the PE.

Spammer's Goal: $\max_{\mathsf{p}} \max\{0, \mathrm{PE}(v;\mathcal{R},\mathsf{p},\mathsf{q})\}$

p: Spamming strategy weights

Defender's Practical Goal

- The defender needs to **minimize** the practical effect
- We combine detector prediction results with the practical effect to formulate a **cost-sensitive loss**

The cost of false negatives

Defender's Goal: $$\min_{\mathsf{q}} \quad \mathcal{L}_{\mathsf{q}} = \frac{1}{|\mathcal{R}(\mathsf{p},\mathsf{q})|} \sum_{r \text{ is FN}} -C_{\text{FN}}(v,r) \log P(y=1|r;\mathsf{q})$$

Detector weights

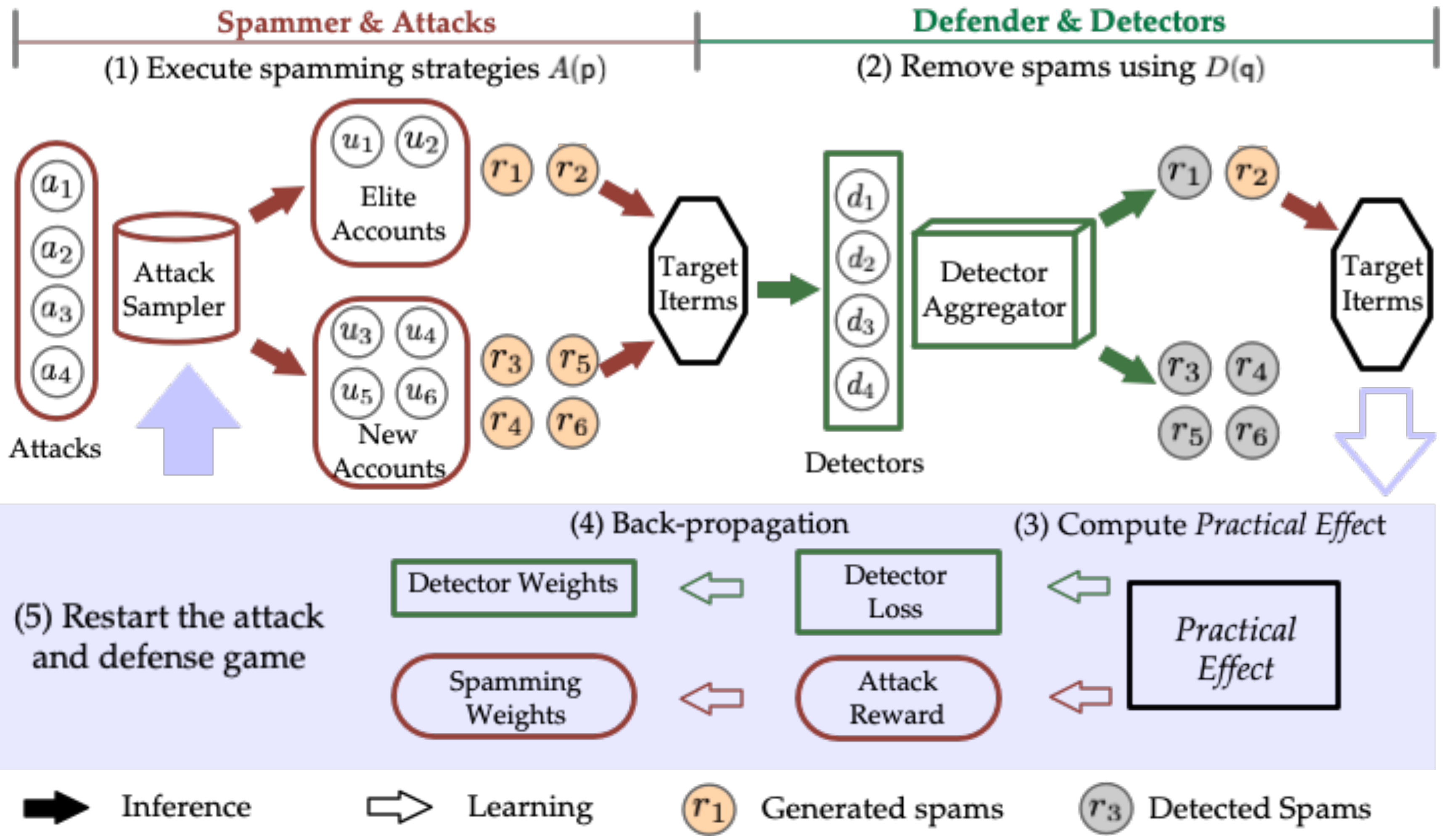
The prediction results of detectors

A Minimax-Game Formulation

Minimax Game Objective: $\min_{\mathbf{q}} \max_{\mathbf{p}} \sum_{v \in \mathcal{V}_T} \max\{0, \mathrm{PE}(v; \mathcal{R}, \mathbf{p}, \mathbf{q})\}$

- The objective function is not differentiable
- Our solution: **multi-agent non-cooperative reinforcement learning** and **SGD optimization**

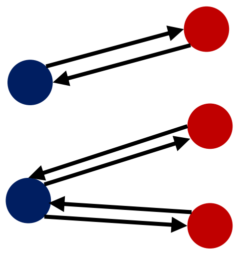
Train a Robust Detector - Nash-Detect



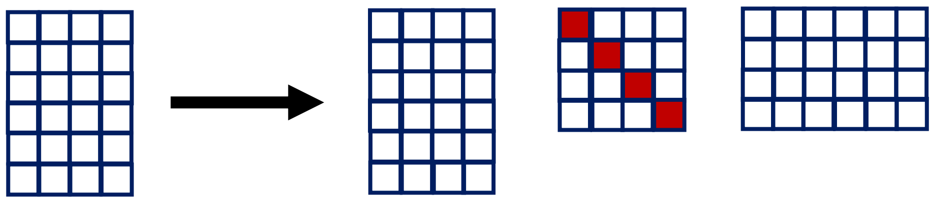
Base Spam Detectors

- **GANG**
- **SpEagle**

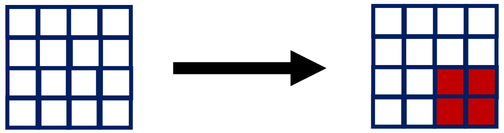
} MRF-based detector



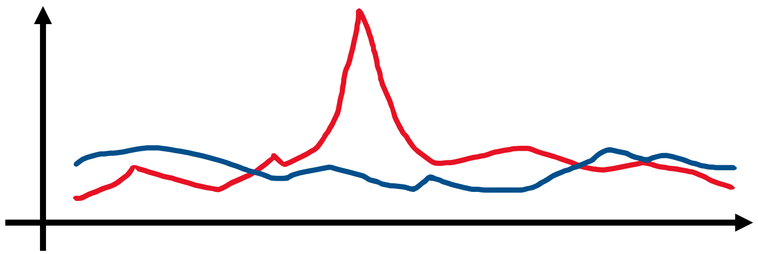
- **fBox** SVD-based detector



- **Fraudar** Dense-block-based detector



- **Prior** Behavior-based detector



Base Spamming Strategies

- **IncBP:** add reviews with minimum suspiciousness based on belief propagation on MRF
- **IncDS:** add reviews with minimum densities on graph composed of accounts, reviews, and products
- **IncPR:** add reviews with minimum prior suspicious scores computed by behavior features
- **Random:** randomly add reviews
- **Singleton:** add reviews with new accounts

Experimental Settings

- Dataset statistics and spamming attack settings

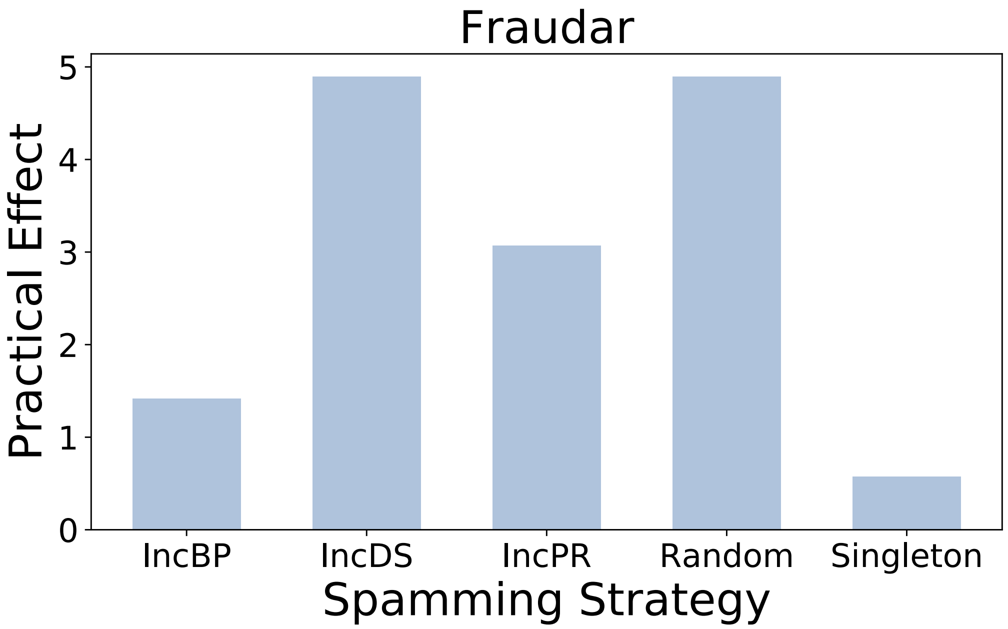
Dataset	# Accounts	# Products	# Reviews	# Controlled elite accounts	# Target products	# Posted fake reviews
YelpChi	38063	201	67395	100	30	450
YelpNYC	160225	923	359052	400	120	1800
YelpZip	260277	5044	608598	700	600	9000

- The spammer controls **elite and new accounts**

- The defender removes **top k** suspicious reviews

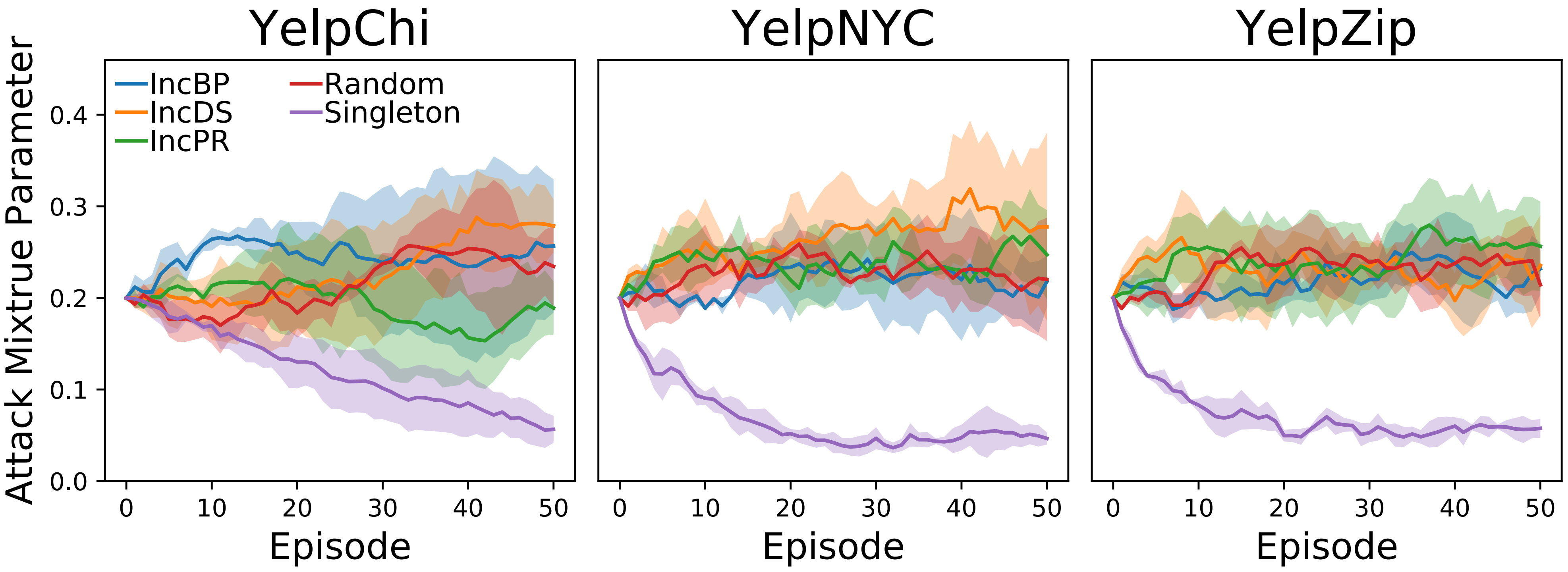
Fixed Detector's Vulnerability

- For a fixed detector (**Fraudar**), the spammer can switch to the spamming strategy with the max practical effect (**IncDS**)



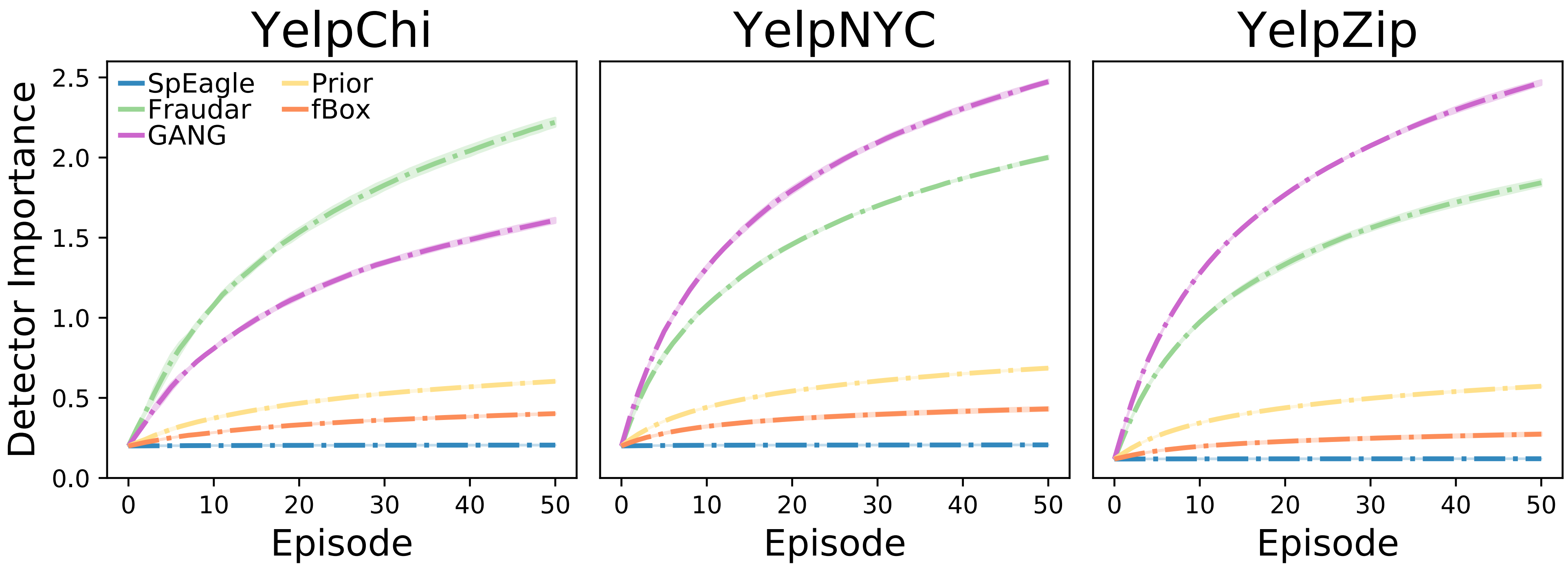
Nash-Detect Training Process

- **Singleton** attack is less effective than other four attacks



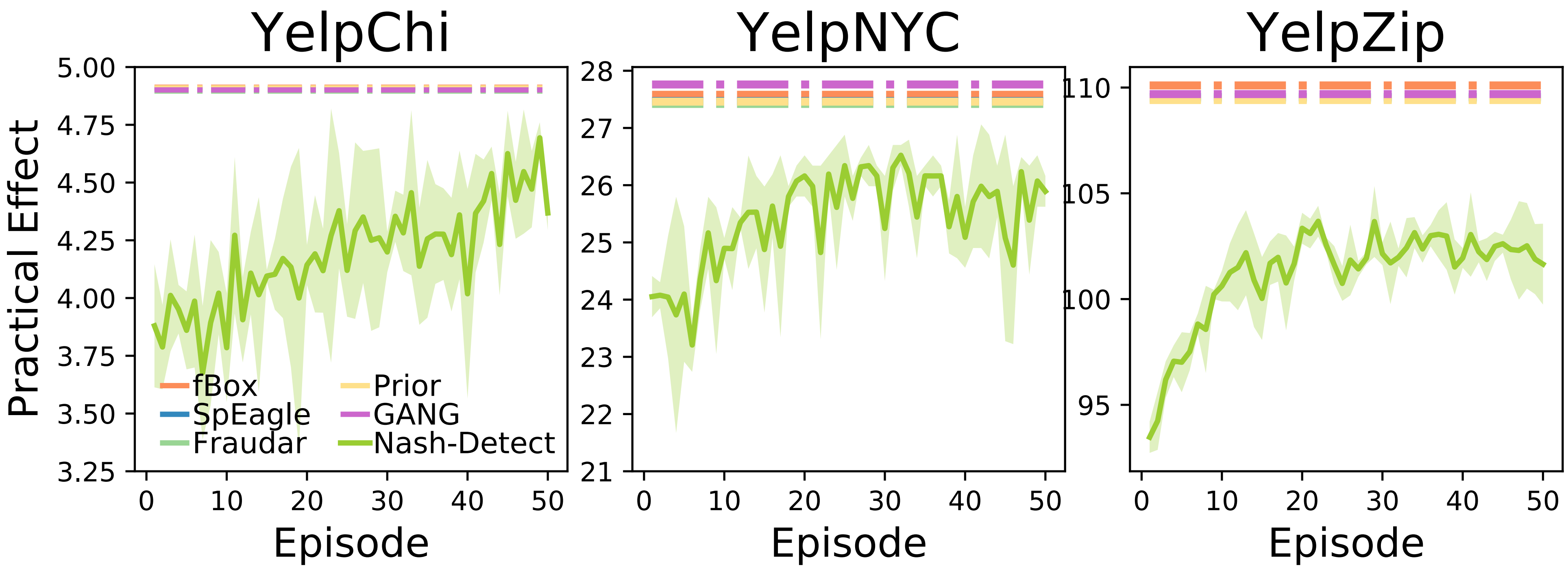
Nash-Detect Training Process

- Nash-Detect can find the optimal detector importance smoothly



Nash-Detect Training Process

- The practical effect of detectors configured by Nash-Detect are always **less than** the worst-case performances



SIGIR'20: Inconsistency Problem

Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection

Zhiwei Liu, Yingtong Dou, Philip S. Yu
Department of Computer Science, University of Illinois at Chicago
{zliu213,ydou5,psyu}@uic.edu

Yutong Deng
School of Software, Beijing University of Posts and Telecommunications
buptdyt@bupt.edu.cn

Hao Peng
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
penghao@act.buaa.edu.cn

Paper: https://arxiv.org/abs/2005.00625
Code: https://github.com/safe-graph/DGFraud/tree/master/algorithms/GraphConsis

CIKM'20: Camouflaging Problem

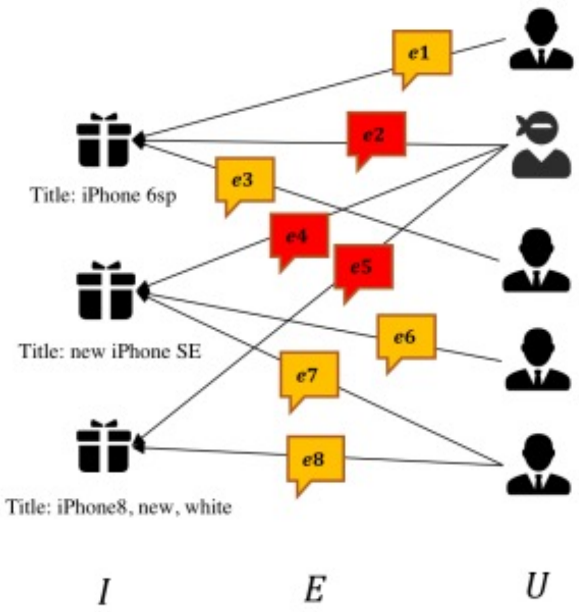
Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters

Yingtong Dou[1], Zhiwei Liu[1], Li Sun[2], Yutong Deng[2], Hao Peng[3], Philip S. Yu[1]
[1]Department of Computer Science, University of Illinois at Chicago
[2]School of Computer Science, Beijing University of Posts and Telecommunications
[3]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
{ydou5,zliu213,psyu}@uic.edu,{l.sun,buptdyt}@bupt.edu.cn,penghao@act.buaa.edu.cn

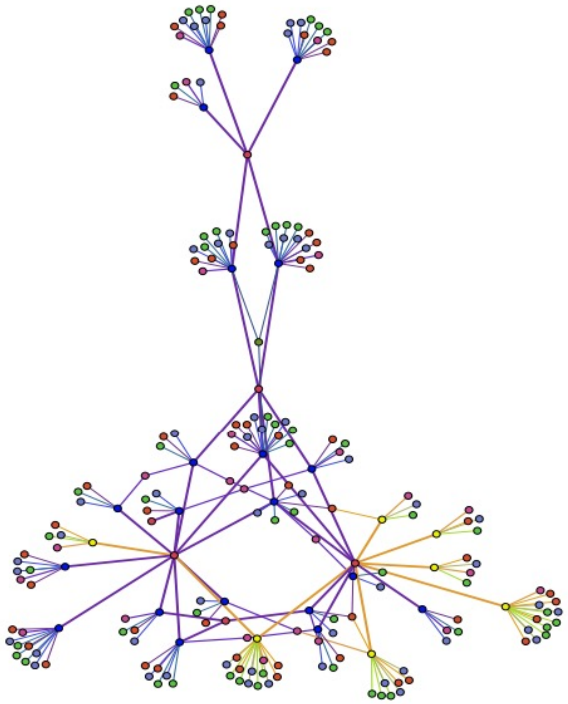
Paper: https://arxiv.org/abs/2008.08692
Code: https://github.com/YingtongDou/CARE-GNN
Improved Model: https://github.com/safe-graph/RioGNN

Graph Models in Industry

Heterogeneous Graphs



User-Review-Item Graph[1]

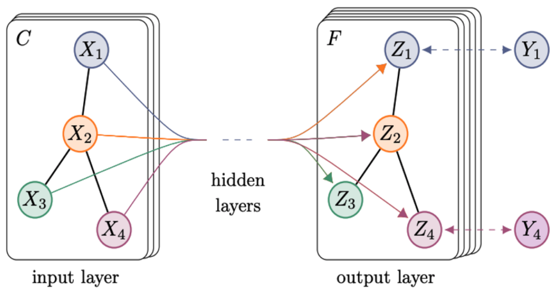


Account-Device Graph[2]

[1 Li, Ao et al. "Spam Review Detection with Graph Convolutional Networks." CIKM (2019)
[2] Liu, Ziqi et al. "Heterogeneous Graph Neural Networks for Malicious Account Detection." CIKM (2018)

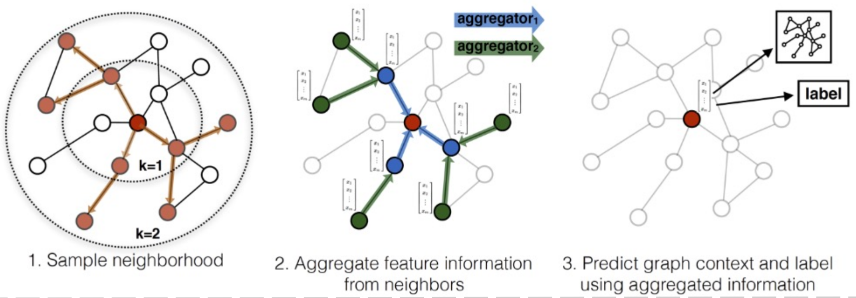
Graph Neural Network

GCN[1]



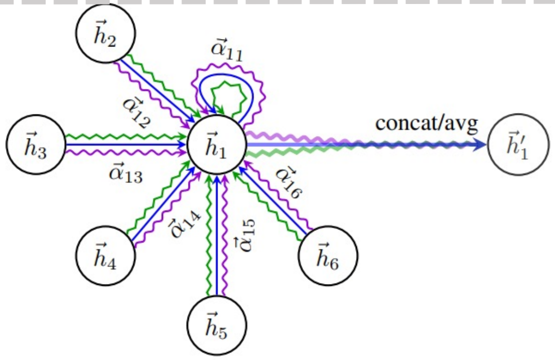
- Directly aggregate neighbors using Laplacian adjacency matrix

GraphSAGE[2]



- Sample and aggregate neighbors

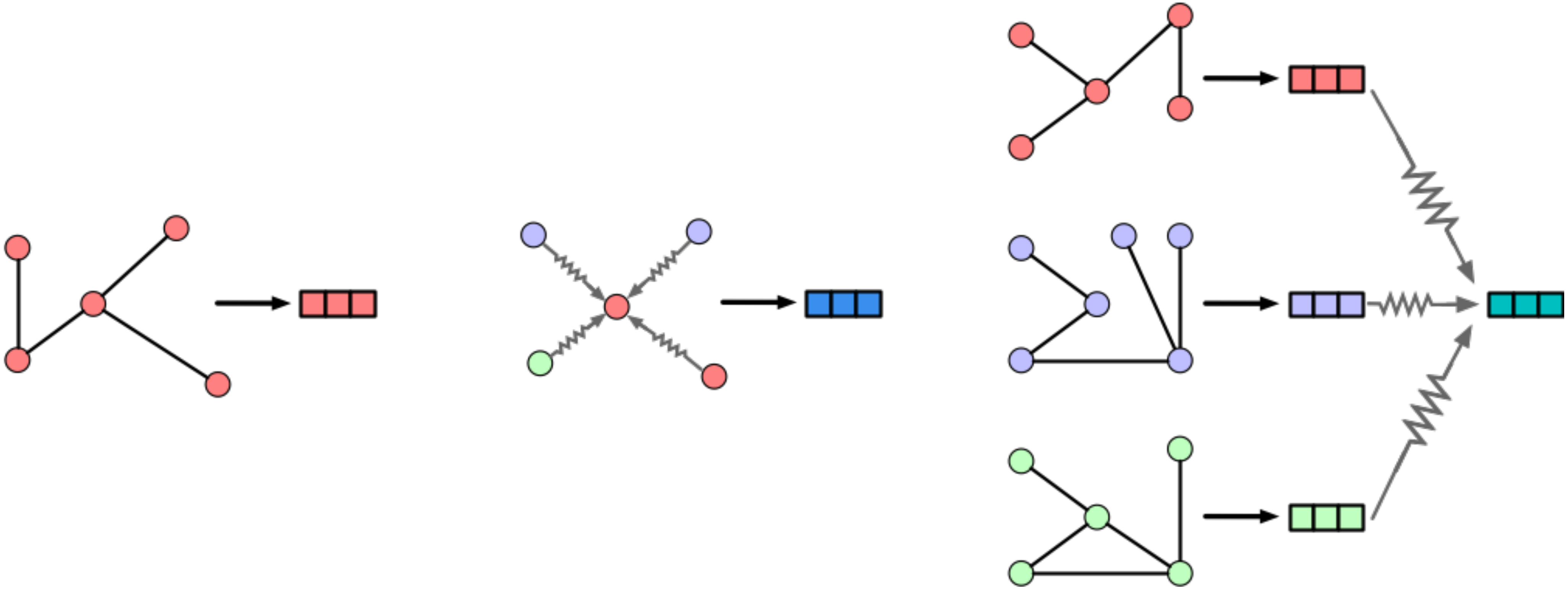
GAT[3]



- Attentively aggregate neighbors

[1] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
[2] W. Hamilton, Hamilton, William L. Ying, Rex Leskovec, Jure. Inductive Representation Learning on Large Graphs , NIPS 2017
[3] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

GNN-based Fraud Detectors



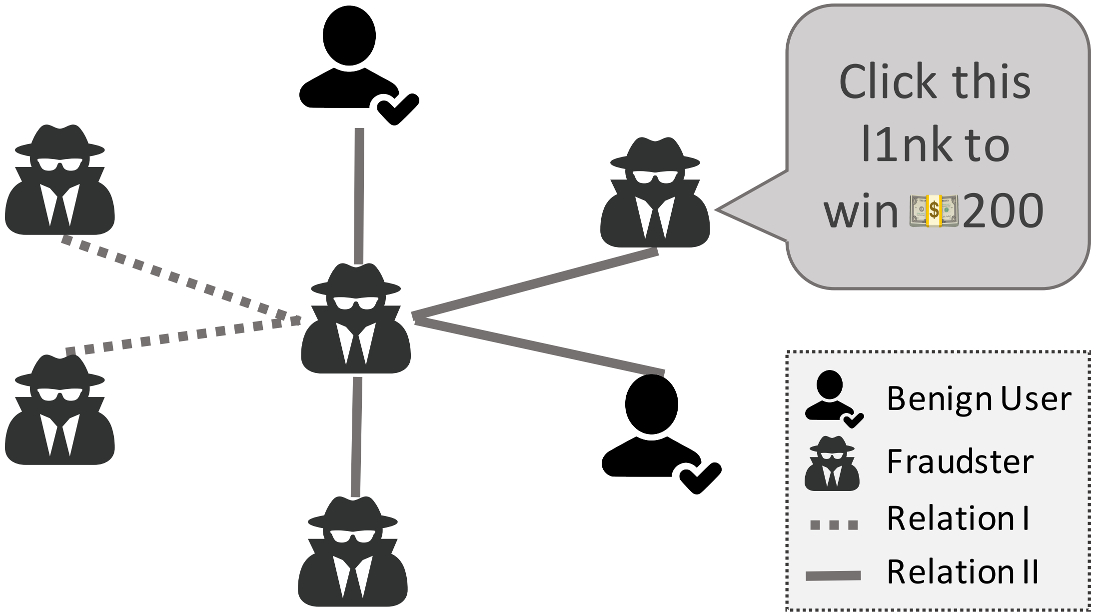
FdGars[1] (GCN-based) GAS[2] (GAT-based) Player2Vec[3] (Hybrid)

[1] Wang, J., Wen, R., Wu, C., Huang, Y. and Xion, J., 2019, May. Fdgars: Fraudster detection via graph convolutional networks in online app review system. WWW 2019.
[2] Li, A., Qin, Z., Liu, R., Yang, Y. and Li, D., 2019, November. Spam review detection with graph convolutional networks. CIKM 2019.
[3] Zhang, Y et, al. November. Key Player Identification in Underground Forums over Attributed Heterogeneous Information Network Embedding Framework. CIKM 2019

Camouflaging Behavior of Fraudsters

- Feature Camouflage

- Relation Camouflage



Principles of Applying GNNs

- The neighboring nodes must be similar
- Only the most informative neighbors are retained
- Each relation should have its importance

Label-aware Similarity Measure

- SIGIR'20 introduces an **unsupervised** similarity measure:

$$s^{(l)}(u,v) = \exp\left(-\|\mathbf{h}_u^{(l)} - \mathbf{h}_v^{(l)}\|_2^2\right)$$

- Unsupervised similarity measure cannot identify **feature camouflage**

- CIKM'20 introduce an **MLP** to encode the label information and use its output as similarity measure:

$$\mathcal{D}^{(l)}(v,v') = \left\|\sigma\left(MLP^{(l)}(\mathbf{h}_v^{(l-1)})\right) - \sigma\left(MLP^{(l)}(\mathbf{h}_{v'}^{(l-1)})\right)\right\|_1$$

Verma, Vikas, et al. "Graphmix: Regularized training of graph neural networks for semi-supervised learning." arXiv preprint arXiv:1909.11715 (2019).
Zhao, Tong, et al. "Error-Bounded Graph Anomaly Loss for GNNs." CIKM 2020.

Similarity-aware Neighbor Selector

- SIGIR'20 uses a neighbor's similarity score among all neighbors as its **sampling** probabilities:

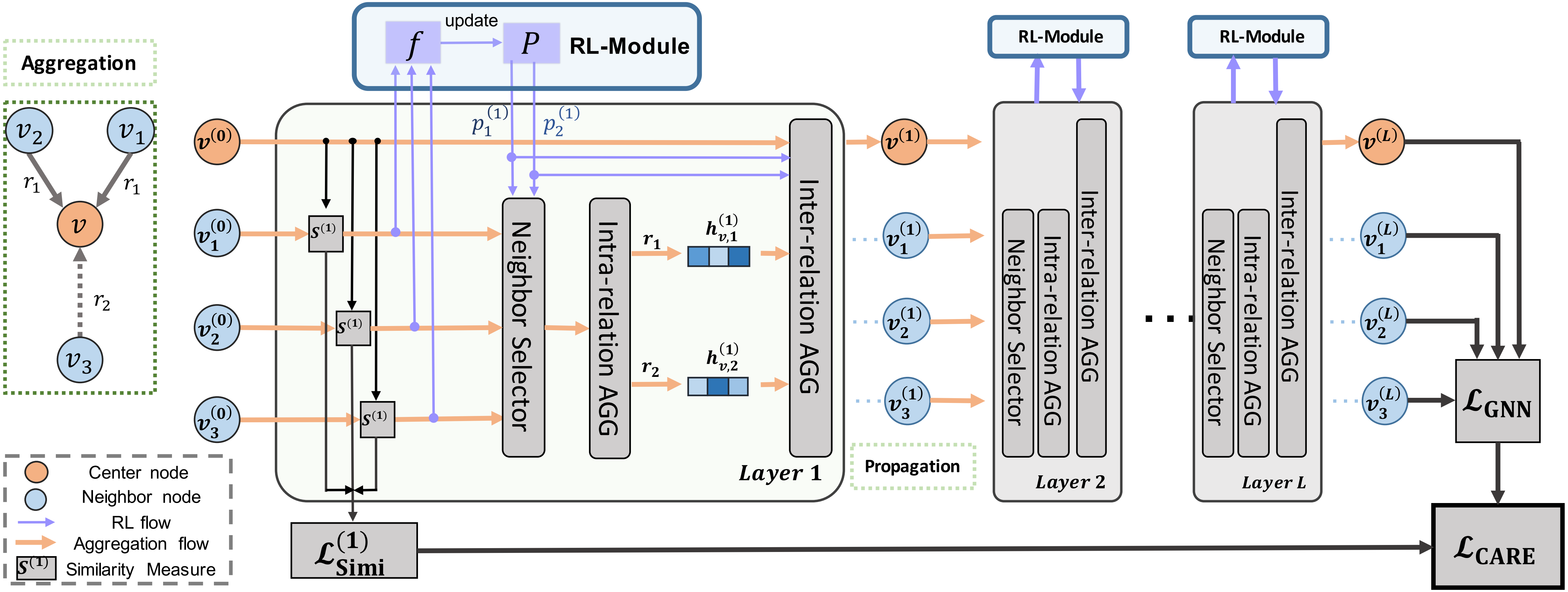
$$p^{(l)}(u;v) = s^{(l)}(u,v) / \sum_{u \in \tilde{\mathcal{N}}_v} s^{(l)}(u,v)$$

- CIKM'20 proposes an adaptive neighbor filtering thresholds using **reinforcement learning** to find the optimal thresholds
- The RL process is a multi-armed bandit with following rules:
  - If the average neighbor similarity score under current epoch is greater than previous epoch, we **increase** the filtering threshold
  - Else, we **decrease** the filtering threshold

Relation-aware Neighbor Aggregator

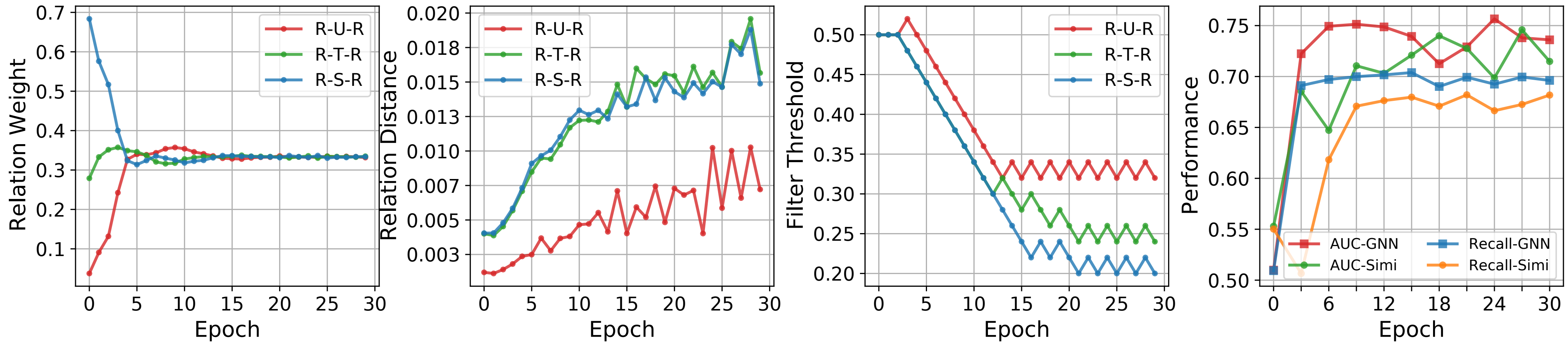
- SIGIR'20 adopts the **attention mechanism** to aggregate neighbors from different relations
- The neighbor filtering threshold of each relation implies the relation importance
- CIKM'20 directly utilize the **neighbor filtering thresholds** as the relation aggregation weights

CARE-GNN Model Overview

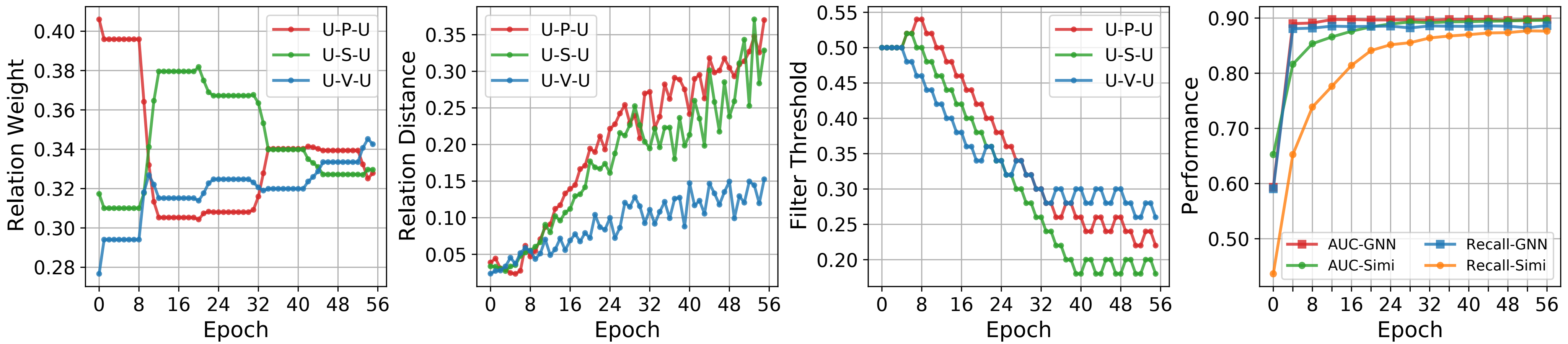


Reinforcement Learning Process

Yelp



Amazon



Overall Evaluation

Table 3: Fraud detection performance (%) on two datasets under different percentage of training data.

	Metric	Train%	GCN	GAT	RGCN	Graph-SAGE	Genie-Path	Player-2Vec	Semi-GNN	Graph-Consis	CARE-*Att*	CARE-*Weight*	CARE-*Mean*	CARE-GNN
Yelp	AUC	5%	54.98	56.23	50.21	53.82	56.33	51.03	53.73	61.58	66.08	71.10	69.83	**71.26**
		10%	50.94	55.45	55.12	54.20	56.29	50.15	51.68	62.07	70.21	71.02	71.85	**73.31**
		20%	53.15	57.69	55.05	56.12	57.32	51.56	51.55	62.31	73.26	74.32	73.32	**74.45**
		40%	52.47	56.24	53.38	54.00	55.91	53.65	51.58	62.07	74.98	74.42	74.77	**75.70**
	Recall	5%	53.12	54.68	50.38	54.25	52.33	50.00	52.28	62.60	63.52	66.64	**68.09**	67.53
		10%	51.10	52.34	51.75	52.23	54.35	50.00	52.57	62.08	67.38	68.35	**68.92**	67.77
		20%	53.87	53.20	50.92	52.69	54.84	50.00	52.16	62.35	68.34	69.07	**69.48**	68.60
		40%	50.81	54.52	50.43	52.86	50.94	50.00	50.59	62.08	71.13	70.22	69.25	**71.92**
Amazon	AUC	5%	74.44	73.89	75.12	70.71	71.56	76.86	70.25	85.46	89.49	89.36	89.35	**89.54**
		10%	75.25	74.55	74.13	73.97	72.23	75.73	76.21	85.29	**89.58**	89.37	89.43	89.44
		20%	75.13	72.10	75.58	73.97	71.89	74.55	73.98	85.50	89.58	**89.68**	89.34	89.45
		40%	74.34	75.16	74.68	75.27	72.65	56.94	70.35	85.50	89.70	89.69	89.52	**89.73**
	Recall	5%	65.54	63.22	64.23	69.09	65.56	50.00	63.29	85.49	88.22	88.31	88.02	**88.34**
		10%	67.81	65.84	67.22	69.36	66.63	50.00	63.32	85.38	87.87	**88.36**	88.12	88.29
		20%	66.15	67.13	65.08	70.30	65.08	50.00	61.28	85.59	88.40	**88.60**	88.00	88.27
		40%	67.45	65.51	67.68	70.16	65.41	50.00	62.89	85.53	88.41	88.45	88.22	**88.48**

Model Advantage

- **Adaptability.** CARE-GNN adaptively selects best neighbors for aggregation given arbitrary multi-relation graph.

- **High-efficiency.** CARE-GNN has a high computational efficiency without attention and deep reinforcement learning.

- **Flexibility.** Many other neural modules and external knowl-edge can be plugged into the CARE-GNN.

SafeGraph (https://github.com/safe-graph)

- **DGFraud**: a GNN-based fraud detection toolbox
  - Ten GNN models developed based on TensorFlow 1.4
- **UGFraud**: an unsupervised graph-based fraud detection toolbox
  - Six classic models, deployed on Pypi
- **GNN-FakeNews**: A collection of GNN-based fake news detectors
  - A benchmark for GNN-based fake news detection based on Twitter data
- Graph-based Fraud Detection Paper List
- Graph Adversarial Learning Paper List

Dataset

- ODDS dataset
  - http://odds.cs.stonybrook.edu/
- Bitcoin dataset
  - https://www.kaggle.com/ellipticco/elliptic-data-set
- Yelp and Amazon
  - https://github.com/YingtongDou/CARE-GNN
- Mobile App Install Fraud
  - https://github.com/mobvistaresearch/CIKM2020-BotSpot

Other Toolbox

- PyOD: A Python Toolbox for Scalable Outlier Detection
  - https://github.com/yzhao062/pyod
- PyODD: An End-to-end Outlier Detection System
  - https://github.com/datamllab/pyodds
- TODS: An Automated Time-series Outlier Detection System
  - https://github.com/datamllab/tods
- Realtime Fraud Detection with GNN on DGL
  - https://github.com/awslabs/realtime-fraud-detection-with-gnn-on-dgl

Other Resources

- Graph Computing for Financial Crime and Fraud Detection Survey
  - https://arxiv.org/abs/2103.03227
- KDD'20 Machine Learning in Finance Workshop
  - https://sites.google.com/view/kdd-mlf-2020/schedule?authuser=0
- KDD'20 Deep Anomaly Detection Tutorial
  - https://sites.google.com/view/kdd2020deepeye/home
- AI for Anti-Money Laundering Blog
  - https://www.markrweber.com/graph-deep-learning
- Awesome Fraud Detection Papers
  - https://github.com/benedekrozemberczki/awesome-fraud-detection-papers

Discussion

- Academic Perspective:
  - The adversarial behavior and robust detector
  - New fraud types, lack of datasets
  - Efficient solvers
  - Model ensemble
  - New learning paradigms
- Industrial Perspective:
  - Fraud vs. Anomaly
  - Sampling is important
  - Cost & return trade off
  - Old but gold[1]
  - Early detection is a challenge

[1] Li, Xiangfeng, et al. "FlowScope: Spotting Money Laundering Based on Graphs." *AAAI*. 2020.

Thanks for listening!

Q & A

Yingtong Dou
University of Illinois at Chicago

Email: ydou5@uic.edu
Twitter: @dozee_sim
Homepage: http://ytongdou.com
Project Page: https://github.com/safe-graph