Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection

Yang Liu[1]; Xiang Ao[1*]; Zidi Qin[1]; Jianfeng Chi[2]; Jinghua Feng[2]; Hao Yang[2]; Qing He[1]

柳 阳[1]; 敖 翔[1*]; 秦紫笛[1]; 池剑锋[2]; 冯景华[2]; 杨 浩[2]; 何 清[1]

1 Institute of Computing Technology, Chinese Academy of Sciences

2 Alibaba Group

* denotes corresponding author.

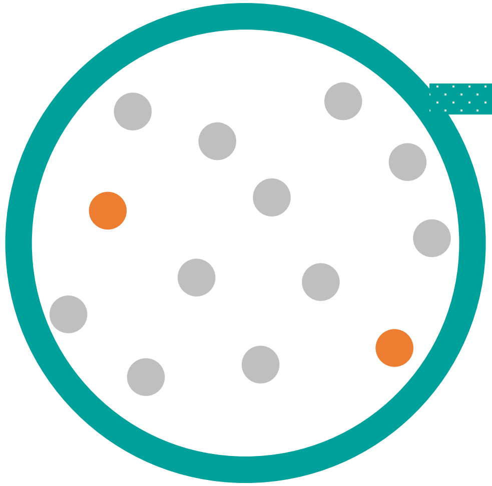
Content

- Background and Motivation
- Method – PC-GNN
- Experiment
- Conclusion and Future Work

Content

- **Background and Motivation**
- Method – PC-GNN
- Experiment
- Conclusion and Future Work

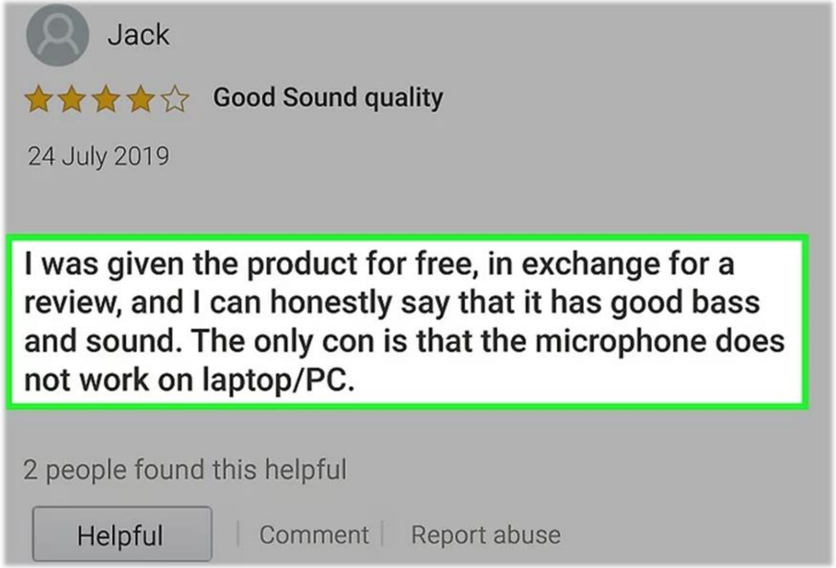
Background



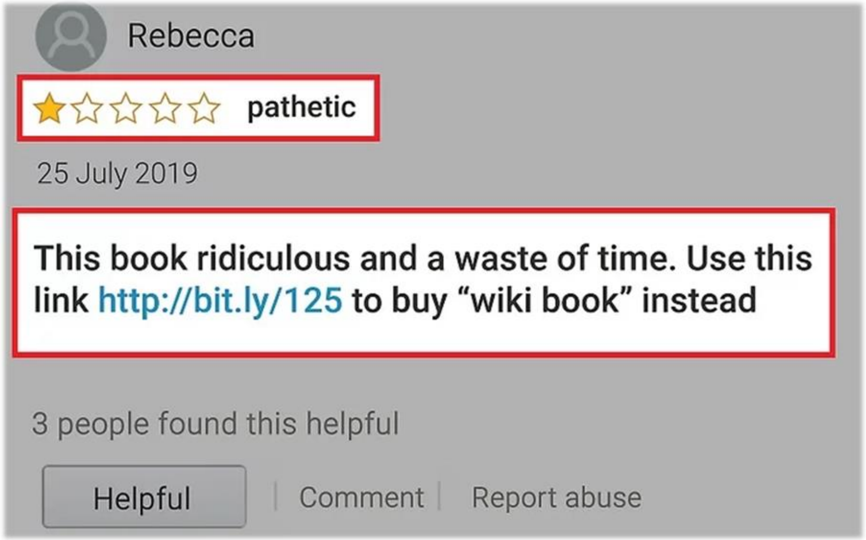
Fraud

- **Opinion fraud (fake/spam review)**
- Financial fraud (fraudster/defaulter)

Online Review Sites



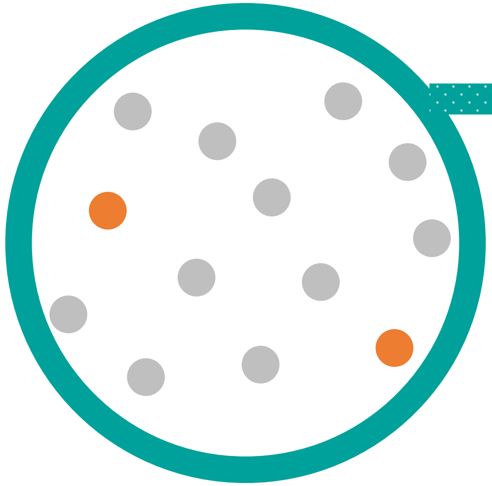
Friend or Paid Reviewer



Rival or Enemy

Images from wikihow, https://www.wikihow.com/Spot-a-Fake-Review-on-Amazon

Background



Fraud

- Opinion fraud (fake/spam review)
- **Financial fraud (fraudster/defaulter)**

Credit Default

Identity Theft

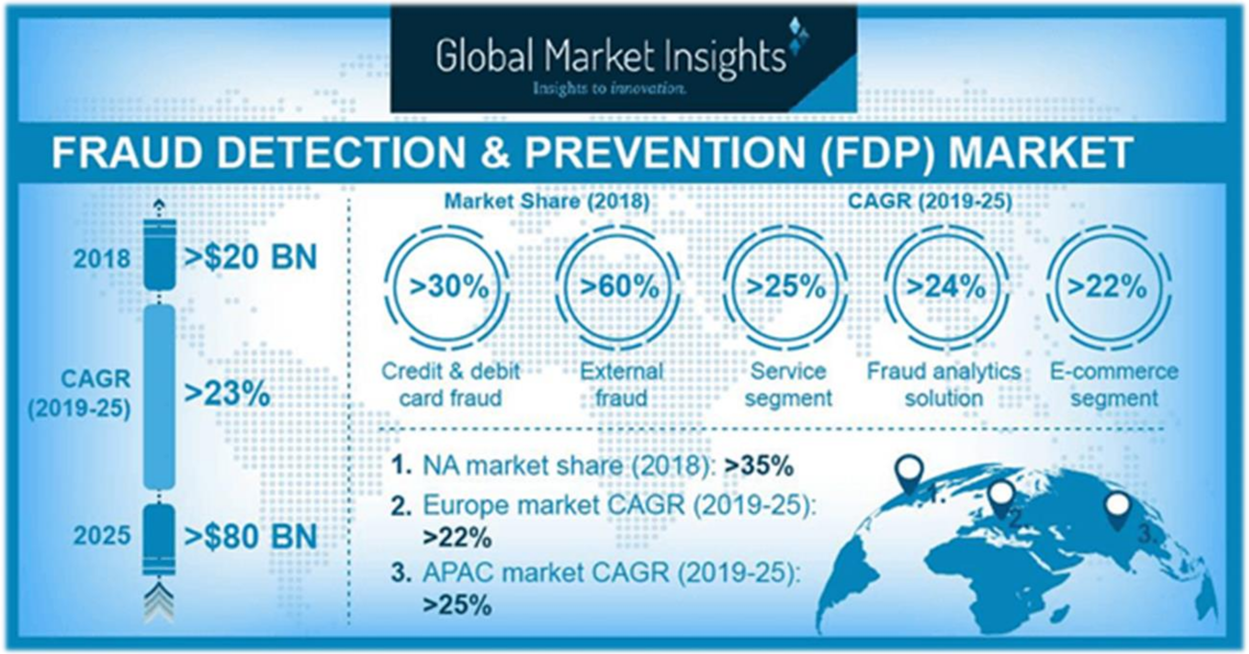
Tax Evasion

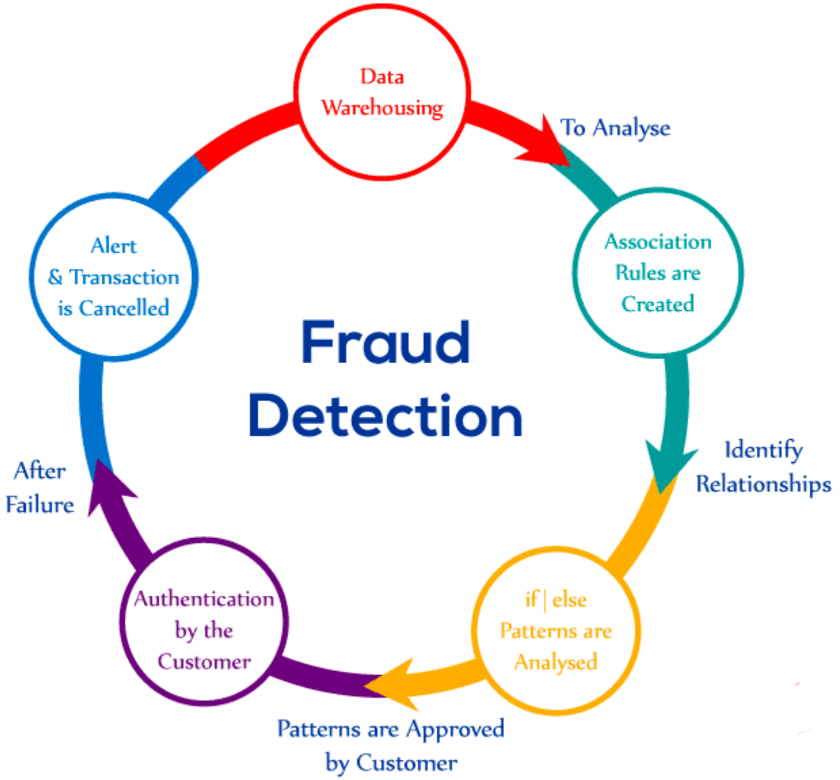
Images from https://hrdailyadvisor.blr.com/2020/04/02/qa-identity-theft-benefits-more-relevant-than-ever/
https://www.icoservices.com/news/reasons-why-doing-illegal-tax-evasion-unnecessary.html

Background

Fraud Detection

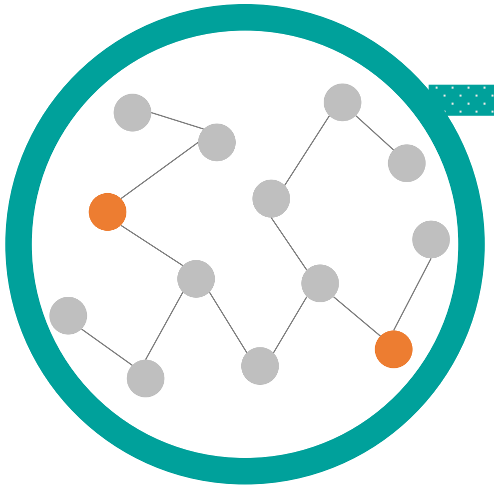
- A set of processes and analyses that allow businesses to identify and prevent unauthorized financial activity.





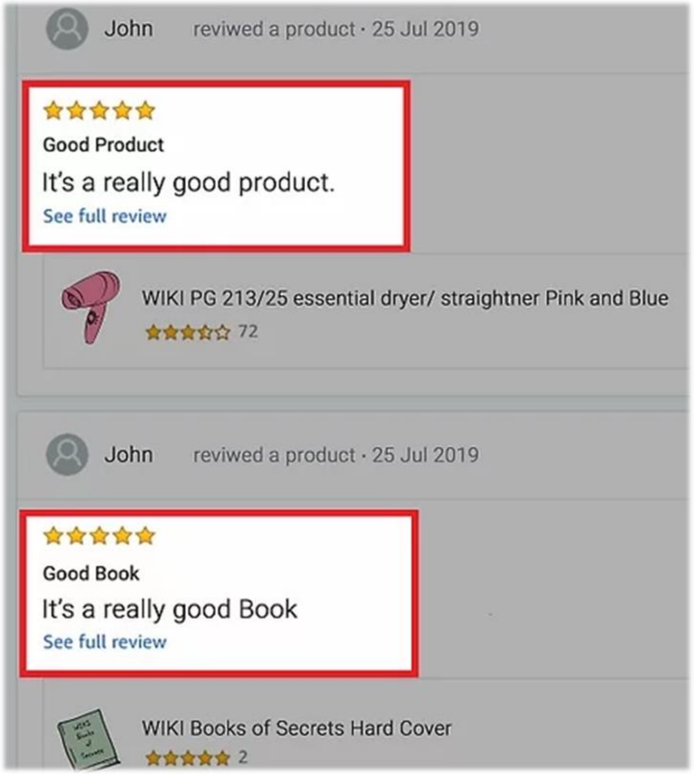
Images from https://www.gminsights.com/industry-analysis/fraud-detection-and-prevention-market https://www.omnisci.com/technical-glossary/fraud-detection-and-prevention

Background



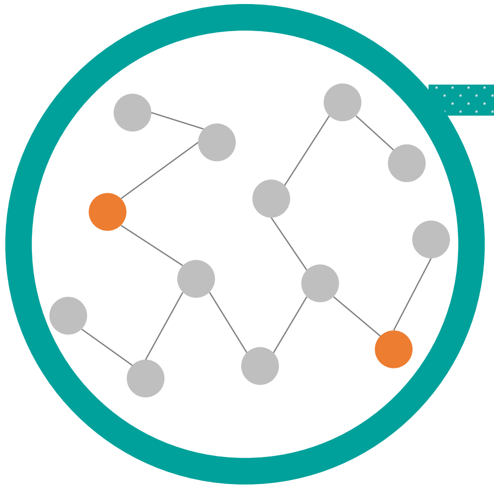
Graph-based Fraud Detection

- Relational data could be modeled as a graph
- Examples: product reviews



- **R-U-R** **R**eviews posted by the same **U**ser
- **R-S-R** **R**eviews under the same product with the same **S**tar rating
- **R-T-R** **R**eviews under the same product in the same mon**T**h

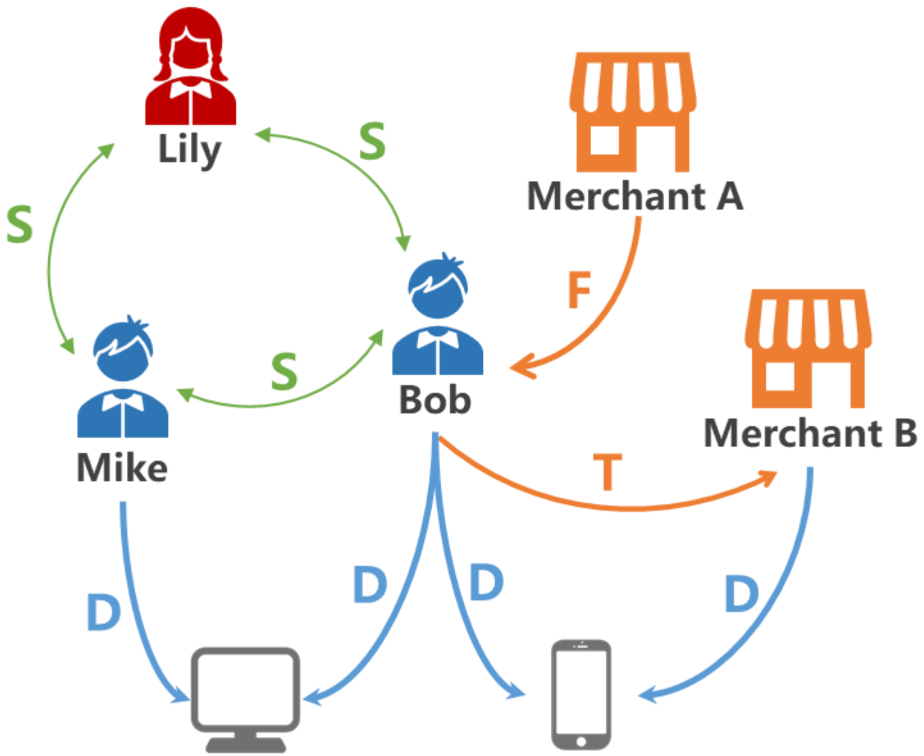
Background



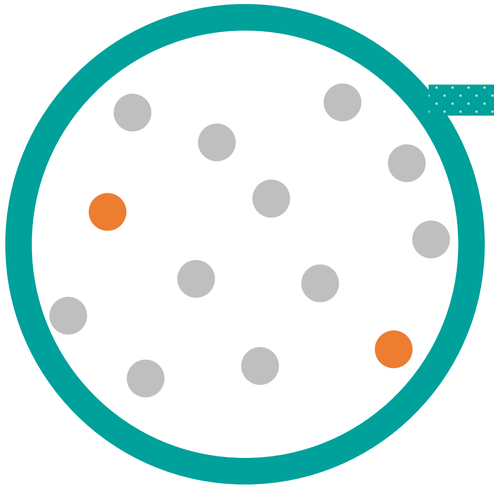
Graph-based Fraud Detection

- Relational data could be modeled as a graph
- Examples: financial scenario

- **U-T-U** **U**ser **T**rades to another
- **U-D-U** **U**sers log in the same **D**evice
- **U-F-U** **U**ser transfer **F**und to another
- **U-S-U** **U**sers have **S**ocial relationships

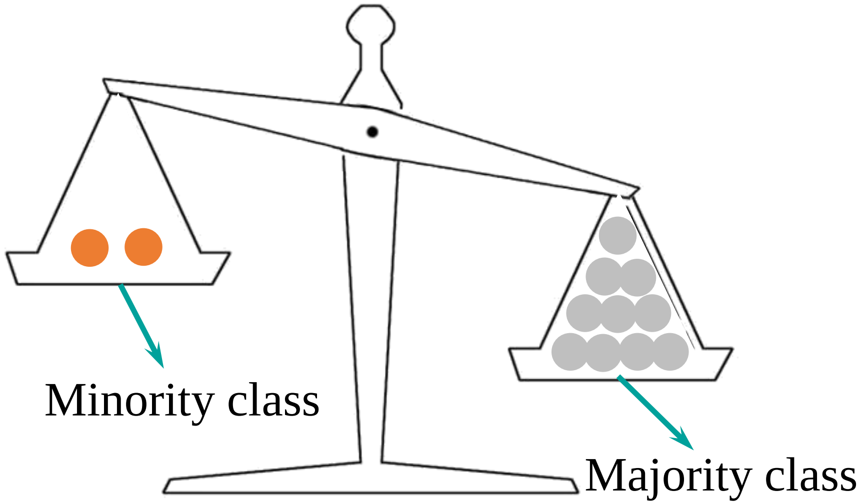


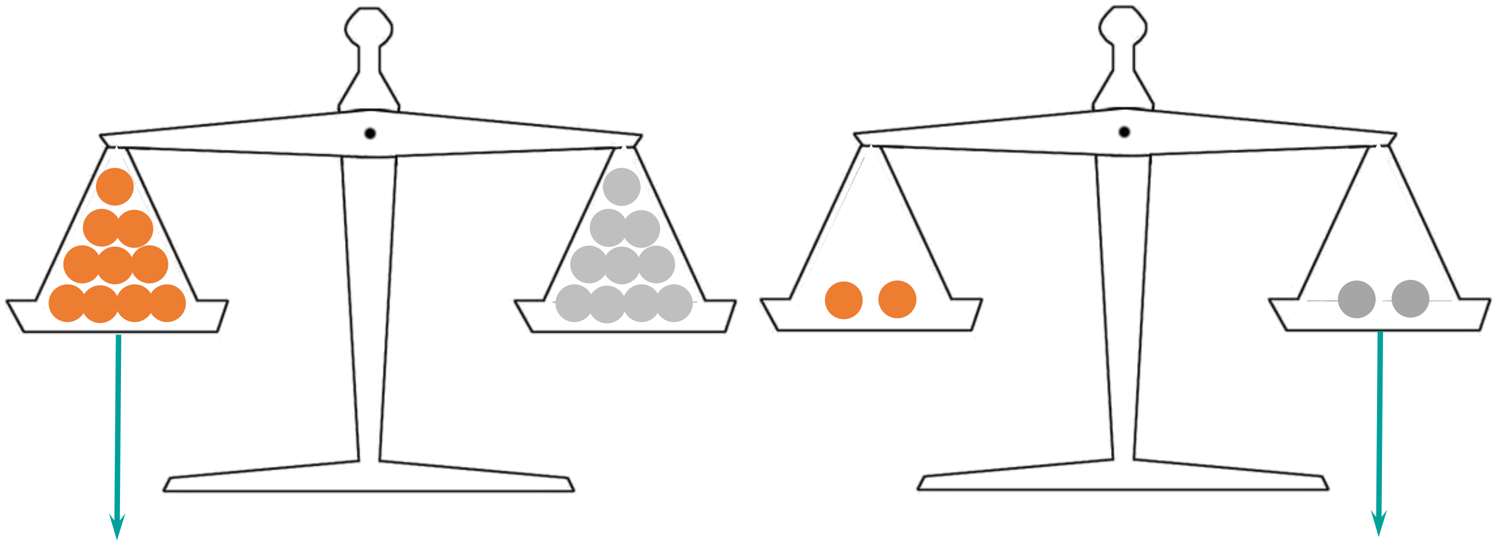
Background



Class imbalance problem

- Only a small fraction of samples belong to the fraud class
- The trained model is easily biased to the majority class



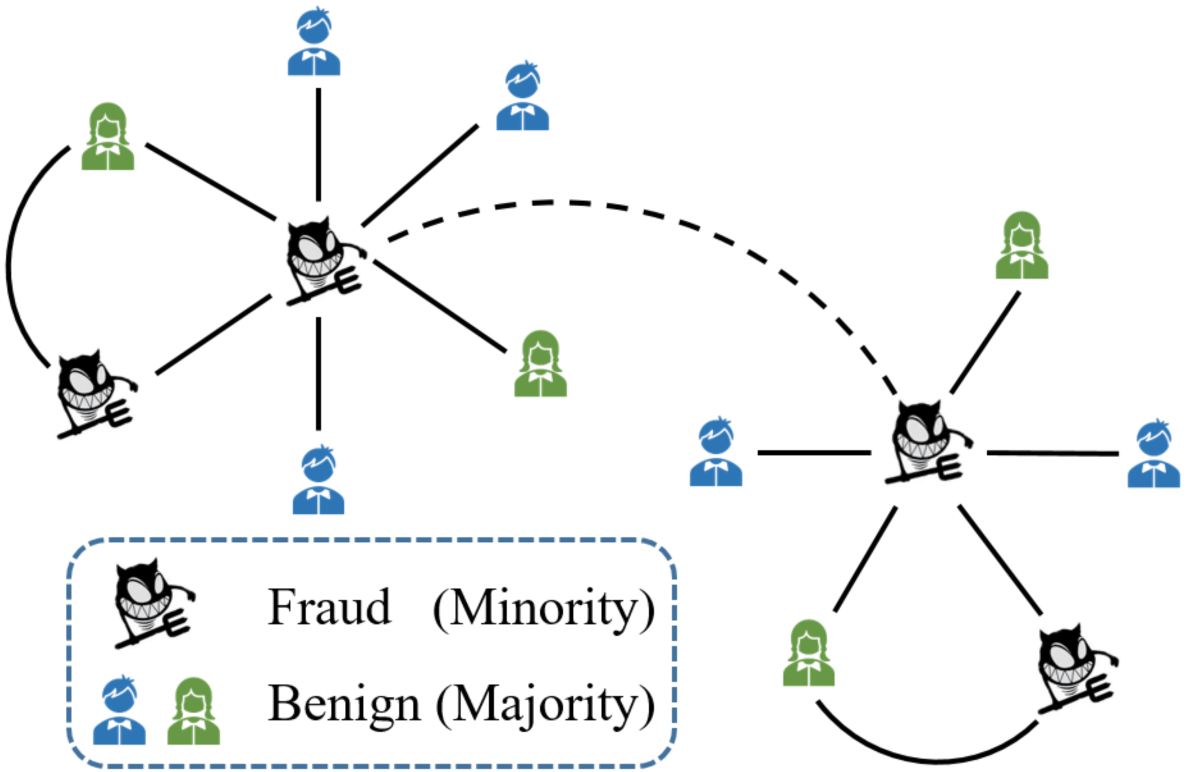


Over-sample the minority class
e.g. SMOTE, ADASYN, etc

Under-sample the majority class
e.g. TU, TRUST, etc.

Motivation

Imbalanced Learning on Graphs



Challenges:

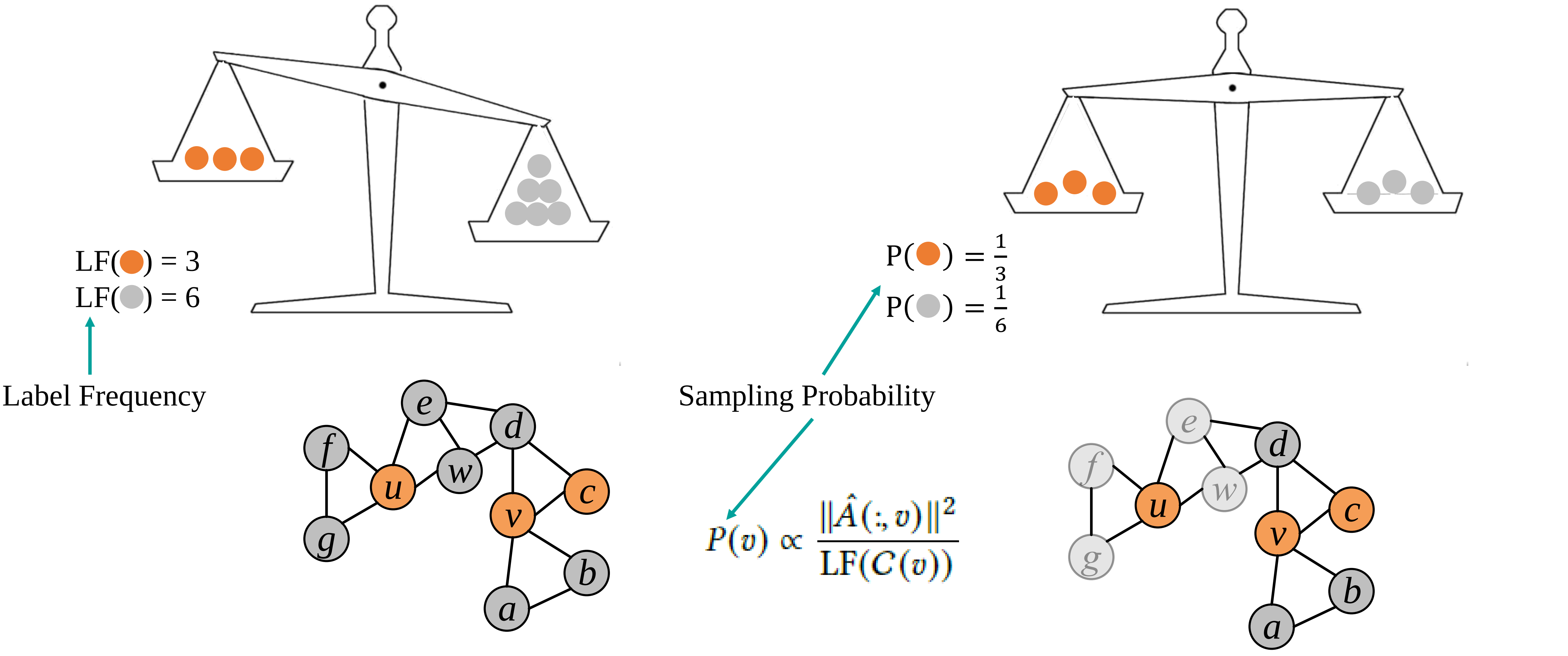
- Camouflage:
  - redundant links between fraudsters and benign users
  - lack necessary links among fraudsters
- Message Aggregation:
  - Most neighbors belong to the majority class
  - The prediction would be biased

Content

- Background and Motivation
- **Method – PC-GNN**
- Experiment
- Conclusion and Future Work

Method

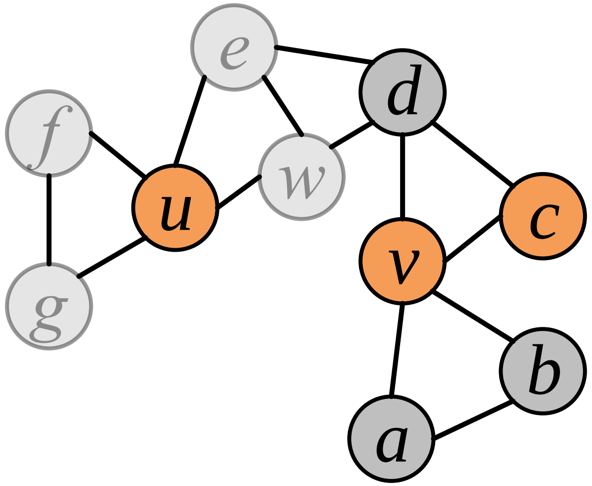
➢ **Pick** nodes from the whole graph

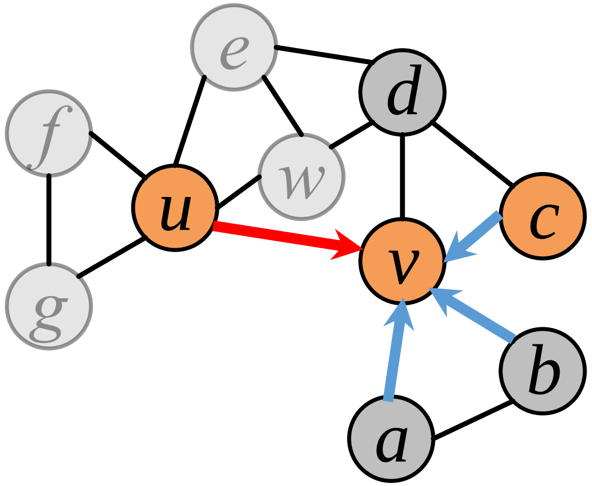


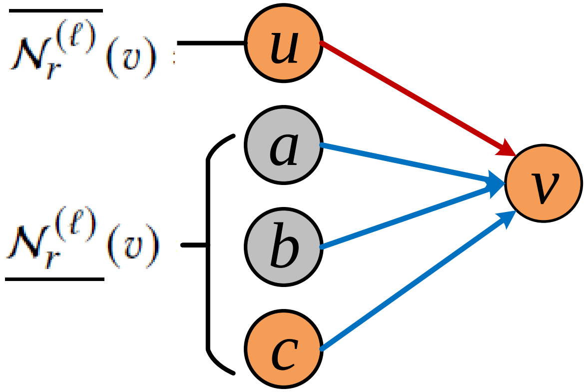
Method

➢ **Choose** neighbors for the minority class

- Over-sample neighbors of the minority class $\overline{\mathcal{N}_r^{(\ell)}}(v) = \left\{u \in \mathcal{V} \,|\, C(u) = C(v) \text{ and } \mathcal{D}_r^{(\ell)}(v,u) < \rho_+\right\}$







- Under-sample neighbors of both classes $\underline{\mathcal{N}_r^{(\ell)}}(v) = \left\{u \in \mathcal{V} \,|\, A_r(v,u) > 0 \text{ and } \mathcal{D}_r^{(\ell)}(v,u) < \rho_-\right\}$

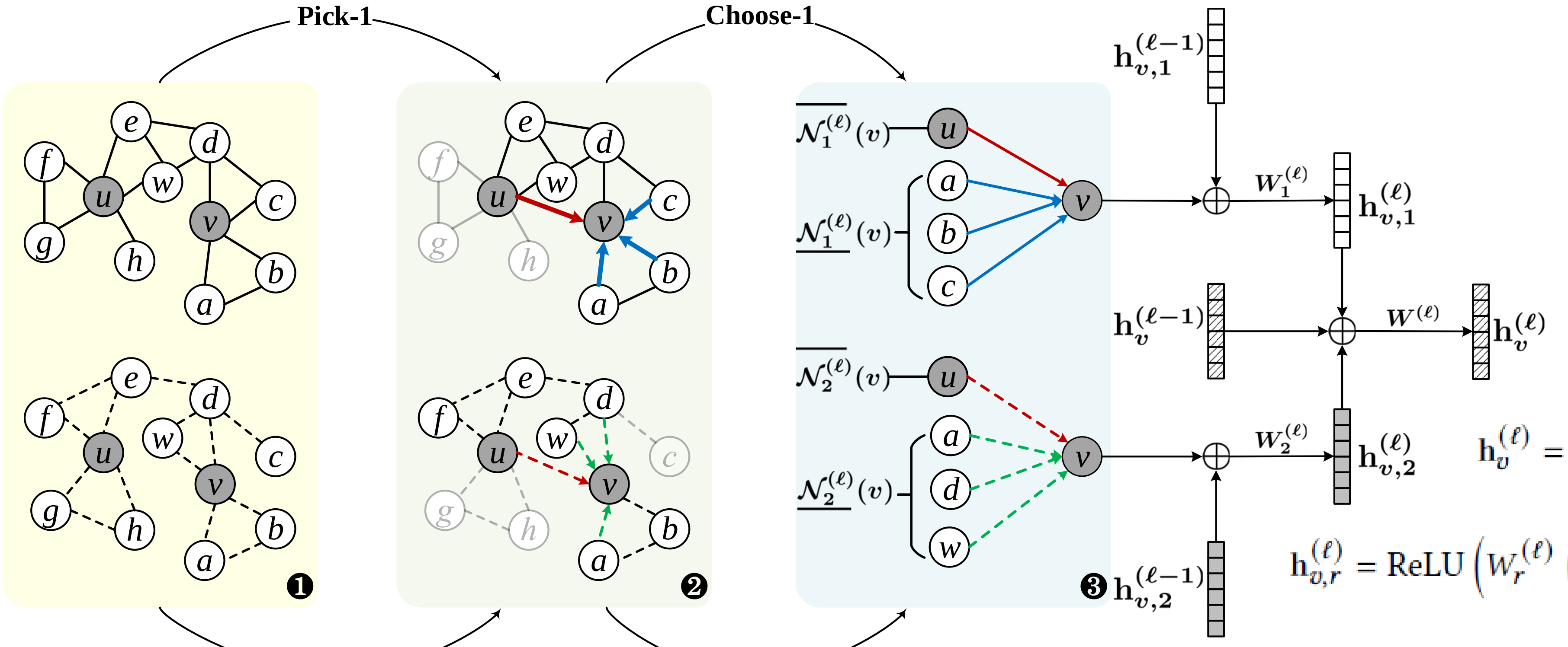
- For minority targets: $\mathcal{N}_r^{(\ell)}(v) = \underline{\mathcal{N}_r^{(\ell)}}(v) \cup \overline{\mathcal{N}_r^{(\ell)}}(v)$
- For majority targets: $\mathcal{N}_r^{(\ell)}(v) = \underline{\mathcal{N}_r^{(\ell)}}(v)$

$$\mathcal{D}_r^{(\ell)}(v,u) = \left\| \mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{v,r}^{(\ell)}\right) - \mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{u,r}^{(\ell)}\right) \right\|_1$$

$$\mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{v,r}^{(\ell)}\right) = \sigma\left(\mathrm{U}_r^{(\ell)} \mathbf{h}_{v,r}^{(\ell)}\right)$$

Method

➢ PC-GNN: Pick and Choose Graph Neural Network



③ Aggregate

$$\mathbf{h}_v^{(\ell)} = \mathrm{ReLU}\left(W^{(\ell)}\left(\mathbf{h}_v^{(\ell-1)} \oplus \mathbf{h}_{v,1}^{(\ell)} \oplus \cdots \oplus \mathbf{h}_{v,R}^{(\ell)}\right)\right)$$

$$\mathbf{h}_{v,r}^{(\ell)} = \mathrm{ReLU}\left(W_r^{(\ell)}\left(\mathbf{h}_{v,r}^{(\ell-1)} \oplus \mathrm{AGG}_r^{(\ell)}\{\mathbf{h}_{u,r}^{(\ell-1)}, u \in \mathcal{N}_r^{(\ell)}(v)\}\right)\right)$$

① Pick $\quad P(v) \propto \dfrac{\|\hat{A}(:,v)\|^2}{\mathrm{LF}(C(v))}$

② Choose

$$\overline{\mathcal{N}_r^{(\ell)}}(v) = \left\{u \in \mathcal{V} \mid C(u) = C(v) \text{ and } \mathcal{D}_r^{(\ell)}(v,u) < \rho_+\right\}$$

$$\underline{\mathcal{N}_r^{(\ell)}}(v) = \left\{u \in \mathcal{V} \mid A_r(v,u) > 0 \text{ and } \mathcal{D}_r^{(\ell)}(v,u) < \rho_-\right\}$$

Fraud
Benign
Relation-1
Relation-2

➢ PC-GNN: Training

- Training the distance function

$$\mathcal{D}_r^{(\ell)}(v,u) = \left\| \mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{v,r}^{(\ell)}\right) - \mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{u,r}^{(\ell)}\right) \right\|_1$$

$$p_{v,r}^{(\ell)} = \mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{v,r}^{(\ell)}\right)$$

$$\mathcal{L}_{\text{dist}} = -\sum_{\ell=1}^{L}\sum_{r=1}^{R}\sum_{v\in\mathcal{V}}\left[y_v \log p_{v,r}^{(\ell)} + (1-y_v)\log\left(1-p_{v,r}^{(\ell)}\right)\right]$$

- Training GNN framework

$$\mathbf{h}_{v,r}^{(\ell)} = \mathrm{ReLU}\left(W_r^{(\ell)}\left(\mathbf{h}_{v,r}^{(\ell-1)} \oplus \mathrm{AGG}_r^{(\ell)}\{\mathbf{h}_{u,r}^{(\ell-1)}, u \in \mathcal{N}_r^{(\ell)}(v)\}\right)\right)$$

$$\mathbf{h}_v^{(\ell)} = \mathrm{ReLU}\left(W^{(\ell)}\left(\mathbf{h}_v^{(\ell-1)} \oplus \mathbf{h}_{v,1}^{(\ell)} \oplus \cdots \oplus \mathbf{h}_{v,R}^{(\ell)}\right)\right)$$

$$p_v = \mathrm{MLP}\left(\mathbf{h}_v^{(L)}\right)$$

$$\mathcal{L}_{\text{gnn}} = -\sum_{v\in\mathcal{V}}\left[y_v \log p_v + (1-y_v)\log(1-p_v)\right]$$

- Overall loss function

$$\mathcal{L} = \mathcal{L}_{\text{gnn}} + \alpha\mathcal{L}_{\text{dist}}$$

Content

- Background and Motivation
- Method – PC-GNN
- **Experiment**
  - RQ1: Does PC-GNN outperform the state-of-the-art methods for graph-based anomaly detection?
  - RQ2: How do the key components benefit the prediction?
  - RQ3: What is the performance with respect to different training parameters?
  - RQ4: If the proposed modules are applied to other GNN models, will it bring performance improvement?
- Conclusion and Future Work

Data

- Public benchmark - Opinion fraud detection
  - **YelpChi**: hotel and restaurant reviews on Yelp
  - **Amazon**: product reviews under the Musical Instrument category

Dataset	#Node	#Edge	IR	Relations	#Relations
YelpChi	45,954	3,846,979	5.9	R-U-R	49,315
				R-S-R	3,402,743
				R-T-R	573,616
Amazon	11,944	4,398,392	13.5	U-P-U	175,608
				U-S-U	3,566,479
				U-V-U	1,036,737

- Real-world dataset - Financial fraud detection
  - Provided by Alibaba Group
  - **M7** collects users from from 2018/07/01 to 2018/07/31
  - **M9** collects users from from 2018/09/01 to 2018/09/30

Dataset	#Node	#Edge	IR	Relations	#Relations
M7	188,673	2,239,344	118.4	U-T-U	2,179,770
				U-D-U	28,630
				U-F-U	24,724
				U-S-U	6,220
M9	253,221	5,568,580	141.5	U-T-U	5,483,056
				U-D-U	40,354
				U-F-U	36,644
				U-S-U	8,526

- Train/Valid/Test:
  - 40%/20%/40%

Experiment

- Compared methods
  - **GCN, GAT:** traditional GNNs
  - **DR-GCN:** dual-regularized GCN for imbalanced classification
  - **GraphSAGE, GraphSAINT:** sampling-based GNNs
  - **GraphConsis, CARE-GNN:** SOTA for graph-based fraud detection
  - **PC-GNN**$_{\backslash P}$**, PC-GNN**$_{\backslash C}$: Model with Pick and Choose removed for ablation study

- Metrics
  - **F1-macro: macro** average of **F1**-score of each class
  - **AUC: A**rea **U**nder the ROC **C**urve
  - **GMean: G**eometric **Mean** of True Positive Rate (TPR) and True Negative Rate (TNR)

Experiment

➢RQ1: Does PC-GNN outperform the state-of-the-art methods for graph-based anomaly detection?

➢Compared with state-of-the-art CARE-GNN$_{\text{[CIKM'20]}}$

- AUC improvement 3.6%~5.2%
- GMean improvement 0.6%~3.7%

Method	Dataset	YelpChi			Amazon		
	Metric	F1-macro	AUC	GMean	F1-macro	AUC	GMean
Baselines	GCN	0.5620±0.0067	0.5983±0.0049	0.4365±0.0262	0.6486±0.0694	0.8369±0.0125	0.5718±0.1951
	GAT	0.4879±0.0230	0.5715±0.0029	0.1659±0.0789	0.6464±0.0387	0.8102±0.0179	0.6675±0.1345
	DR-GCN	0.5523±0.0231	0.5921±0.0195	0.4038±0.0742	0.6488±0.0364	0.8295±0.0079	0.5357±0.1077
	GraphSAGE	0.4405±0.1066	0.5439±0.0025	0.2589±0.1864	0.6416±0.0079	0.7589±0.0046	0.5949±0.0349
	GraphSAINT	0.5960±0.0038	0.6999±0.0029	0.5908±0.0298	0.7626±0.0032	0.8701±0.0025	0.7963±0.0091
	GraphConsis	0.5870±0.0200	0.6983±0.0302	0.5857±0.0385	0.7512±0.0325	0.8741±0.0334	0.7677±0.0486
	CARE-GNN	0.6332±0.0094	0.7619±0.0292	0.6791±0.0359	0.8990±0.0073	0.9067±0.1115	0.8962±0.0018
Ablation	PC-GNN$_{\setminus P}$	0.5136±0.0147	0.7844±0.0013	0.2336±0.0356	**0.9158±0.0024**	0.9469±0.0018	0.8782±0.0068
	PC-GNN$_{\setminus C}$	**0.6634±0.0058**	0.7847±0.0021	0.6258±0.0378	0.8929±0.0171	0.9529±0.0035	0.9006±0.0045
Ours	PC-GNN	0.6300±0.0230	**0.7987±0.0014**	**0.7160±0.0130**	0.8956±0.0077	**0.9586±0.0014**	**0.9030±0.0044**

Experiment

➢ Compared with state-of-the-art CARE-GNN$_{[CIKM'20]}$

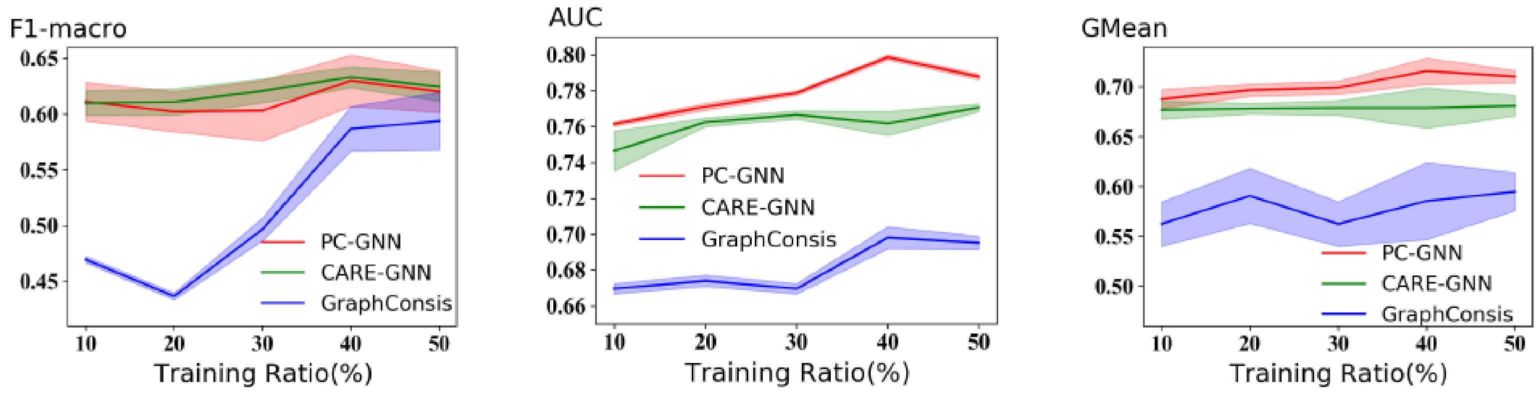
- AUC improvement 2.6%~3.5%
- GMean improvement 28.4%~31.9%

Method	Dataset	M7			M9		
	Metric	F1-macro	AUC	GMean	F1-macro	AUC	GMean
Baselines	GCN	0.3108±0.0256	0.6107±0.0041	0.5456±0.0159	0.3016±0.0574	0.5790±0.0040	0.5241±0.0422
	GAT	0.2746±0.0168	0.6083±0.0149	0.5016±0.0168	0.2698±0.0069	0.5647±0.0069	0.4354±0.0346
	DR-GCN	0.3070±0.0232	0.7195±0.0208	0.5647±0.0403	0.5055±0.0012	0.6637±0.0236	0.3106±0.0417
	GraphSAGE	0.5186±0.0030	0.6790±0.0029	0.1605±0.0132	0.5020±0.0026	0.6342±0.0040	0.0525±0.0362
	GraphSAINT	0.5149±0.0036	0.6915±0.0068	0.2547±0.0459	0.5018±0.0019	0.6587±0.0049	0.1864±0.0354
	GraphConsis	0.5236±0.0087	0.6826±0.0049	0.2734±0.0548	0.5124±0.0043	0.6743±0.0076	0.2302±0.0467
	CARE-GNN	0.5578±0.0015	0.7836±0.0020	0.3451±0.0098	0.5361±0.0035	0.7579±0.0060	0.2908±0.0294
Ablation	PC-GNN$_{\backslash P}$	0.4979±0.0000	0.7434±0.0042	0.0000±0.0000	0.4982±0.0000	0.6575±0.0069	0.0000±0.0000
	PC-GNN$_{\backslash C}$	0.5735±0.0017	0.8132±0.0031	0.4362±0.0254	0.5353±0.0028	0.7668±0.0038	0.3138±0.0387
Ours	PC-GNN	**0.5749±0.0044**	**0.8192±0.0032**	**0.6645±0.0422**	**0.5370±0.0021**	**0.7847±0.0019**	**0.5740±0.0391**

➢RQ2: How do the key components benefit the prediction?

Experiment

➢RQ3: What is the performance with respect to different training parameters?



(a) F1-macro (b) AUC (c) GMean

➢RQ4: If the proposed modules are applied to other GNN models, will it bring performance improvement?

Dataset	YelpChi			Amazon		
Metric	F1-macro	AUC	GMean	F1-macro	AUC	GMean
GCN	0.4645±0.0033	0.5983±0.0049	0.0483±0.0373	0.5297±0.0539	0.8369±0.0125	0.1904±0.1716
GCN(T)	**0.5620±0.0067**	0.5983±0.0049	0.4365±0.0262	0.6486±0.0694	0.8369±0.0125	0.5718±0.1951
GCN(P)	0.5540±0.0275	**0.6122±0.0643**	**0.5114±0.0107**	**0.7138±0.0086**	**0.8773±0.0027**	**0.7823±0.0374**
GraphSAGE	0.4608±0.0000	0.5439±0.0025	0.0000±0.0000	0.4751±0.0000	0.7589±0.0046	0.0000±0.0000
GraphSAGE(T)	0.4405±0.1066	0.5439±0.0025	0.2589±0.1864	**0.6416±0.0079**	0.7589±0.0046	0.5949±0.0349
GraphSAGE(P)	**0.6178±0.0286**	**0.7765±0.0025**	**0.6952±0.0176**	0.5831±0.0227	**0.7627±0.0097**	**0.6993±0.0081**

Content

- Background and Motivation
- Method – PC-GNN
- Experiment
- **Conclusion and Future Work**

Conclusion and Future work

THE WEB CONFERENCE

- Conclusion
  - We propose a GNN-based imbalanced learning method named PC-GNN to solve the class imbalance problem in graph-based fraud detection
  - Experiments on two benchmark opinion fraud datasets and two real-world financial fraud datasets demonstrate the effectiveness of the proposed framework.

- Future Work
  - Graph structure learning for imbalanced graph data

Thanks for listening!

If you have any question, feel free to contact us at

liuyang17z@ict.ac.cn
aoxiang@ict.ac.cn

Paper and slides are available at
https://ponderly.github.io/