Fraud Dectection

Group 11

Nguyễn Như Quỳnh – 30%
Nguyễn Minh Anh – 30%
Nguyễn Thị Hoài Linh – 20%
Lê Trung Hiếu – 20%

Table of Contents

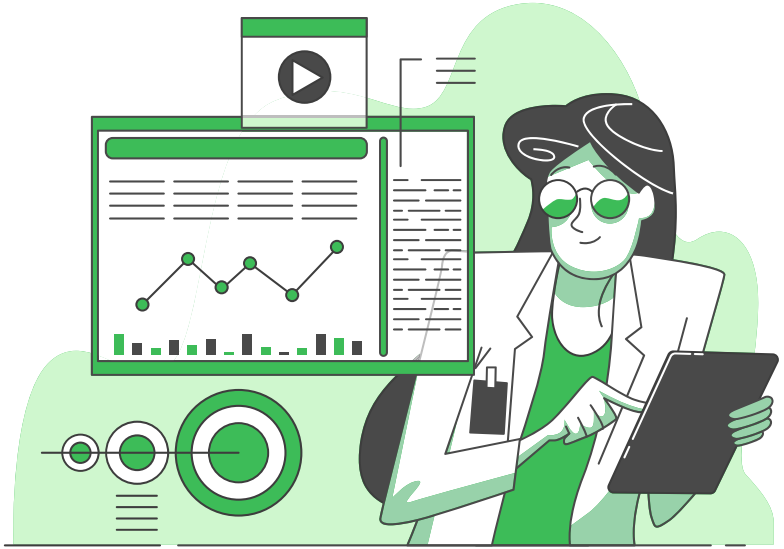
01 Introduction

02 Data Visualization

03 Models

04 Training Model

05 Results



Introduction

Background

Fraud

Opinion fraud: fake/spam review

Online Review Sites

Friend or Paid Reviewer

messihihi

★★★★☆

Reviewed in the United States on November 23, 2022

Size: 10 | Color: Army Green | Verified Purchase

Đôi giày này đi quá là sướng. Mình chân to như voi còn đi cực thoải mái và êm chân. Mọi người nên mua nhé. Ship hàng nhanh như cơn gió, đóng gói hàng cẩn thận kinh khủng, lấy búa bậy mãi còn không ra. Cảm ơn shop iu <3

One person found this helpful

Helpful | Report abuse

Rival or Enemy

messihaha

★☆☆☆☆

Reviewed in the United States on November 23, 2022

Size: 10 | Color: Army Green | Verified Purchase

Đôi giày này quá đểu, vừa đi vào chân đã bục ra lòi cả ngón chân ra ngoài. Mọi người muốn mua giày xịn giống mẫu mã này truy cập vào link shop này nhé mọi người https://www.amazon.com/dp/B08KY6GGB5

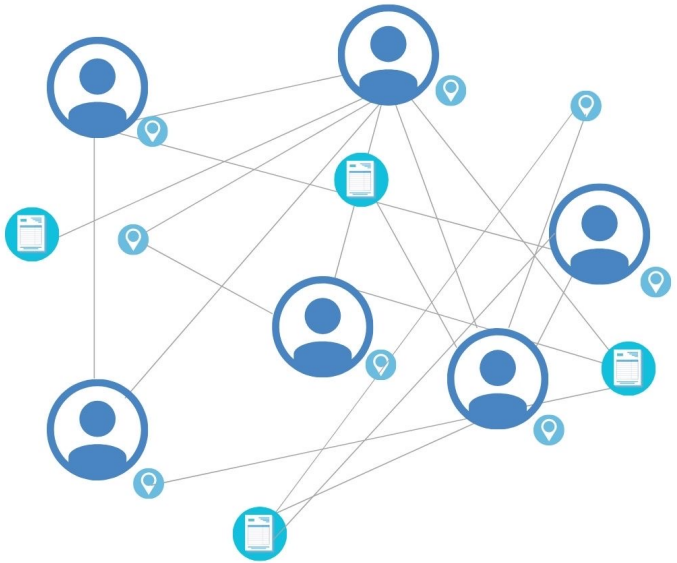
One person found this helpful

Helpful | Report abuse

Background

Graph-based Fraud Detection

Relational data could be modeled as a graph



Task: binary classification

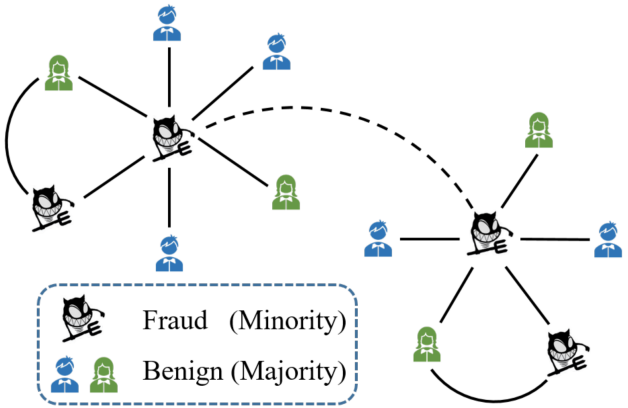
Input:
nodes included features of reviewers, products...,
edges as relationship between nodes

Output: 0 or 1 - fraud or not

Dataset

Amazon Dataset

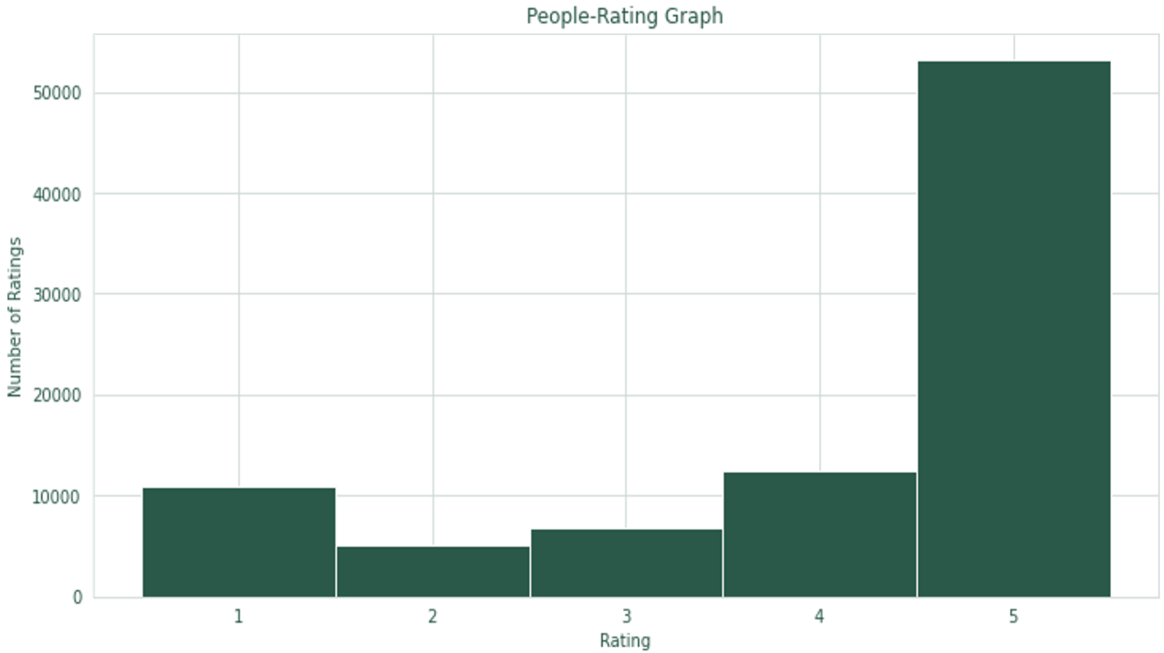
- Our **Amazon dataset** is a subset of Amazon's product dataset [1]. It contains more than **34,000** consumer reviews.
- We extracted **11,949** product reviews under the **musical instrument** category.
- Users with useful voting rates **greater than 80%** as **benign** entities, users with useful voting rates **less than 20%** as **fraudulent** entities.



Data Visualization

Data Visualization

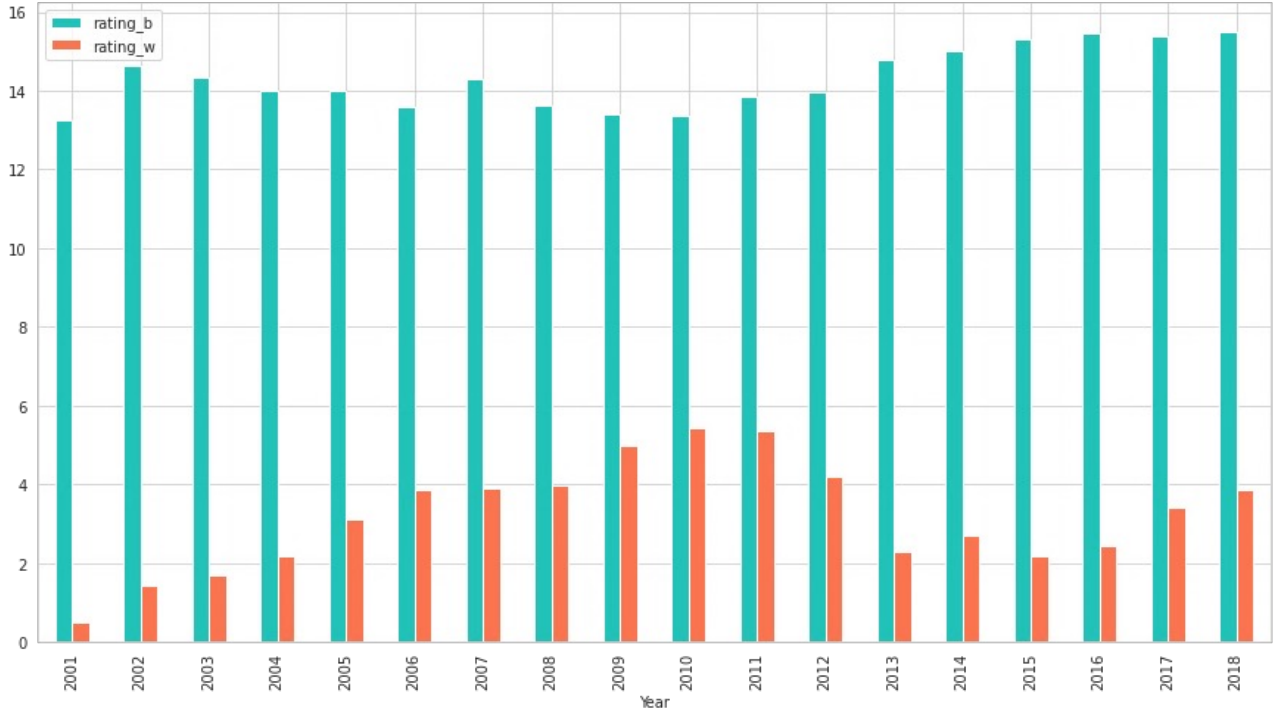
General Analysis



- The histogram shows the number of reviews for each rating on Amazon.
- The highest rating indicates customer satisfaction on the product of 5 and also has the highest number of reviewers (more than 50,000 reviews).
- However, there are more than 10000 reviews that are 1 expressing dissatisfaction.

Data Visualization

General Analysis



The rate of 5 stars rating each year has not changed too much. Each year accounts for about 13-15.5% of the total reviews.

During the period 2006-2012, 1 star rating accounted for about 5% per year then gradually decreased.

During periods of increased 1-star rating, 5-star ratings showed a slight decrease.

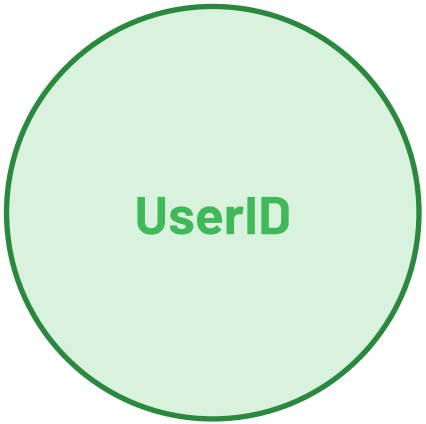
Graph Visualization

Graph Construction

Nodes: UserID

Node feature selection: 25 manual features including

- the number of rated products
- the length of the user name
- the number and ratio of each rating level given by the user
- the ratio of positive and negative reviews
- the user's rating
- the total number of useful and useless votes obtained by the user
- the ratio of useful votes and useless votes
- number of days between the user's first and last rating
- same date indicator
- comment text sentiment
- etc.

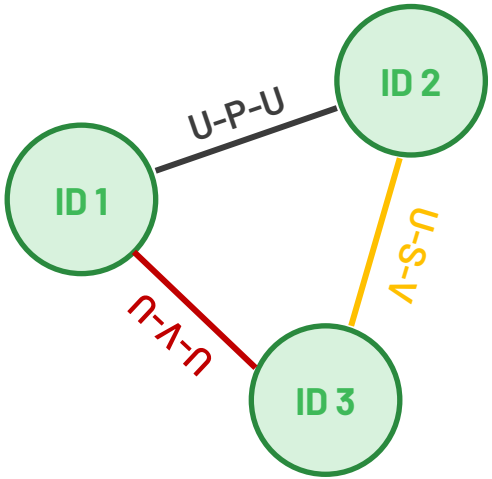


Graph Visualization

Graph Construction

Relationships: Three kinds of relations for the multi-relational graph

1) **U-P-U:** it connects users reviewing at least one same product
2) **U-S-U:** it connects users having at least one same star rating within one week
3) **U-V-U:** it connects users with top 5% mutual review text similarities (measured by TF-IDF) among all users



Graph Visualization

Dataset Statistics

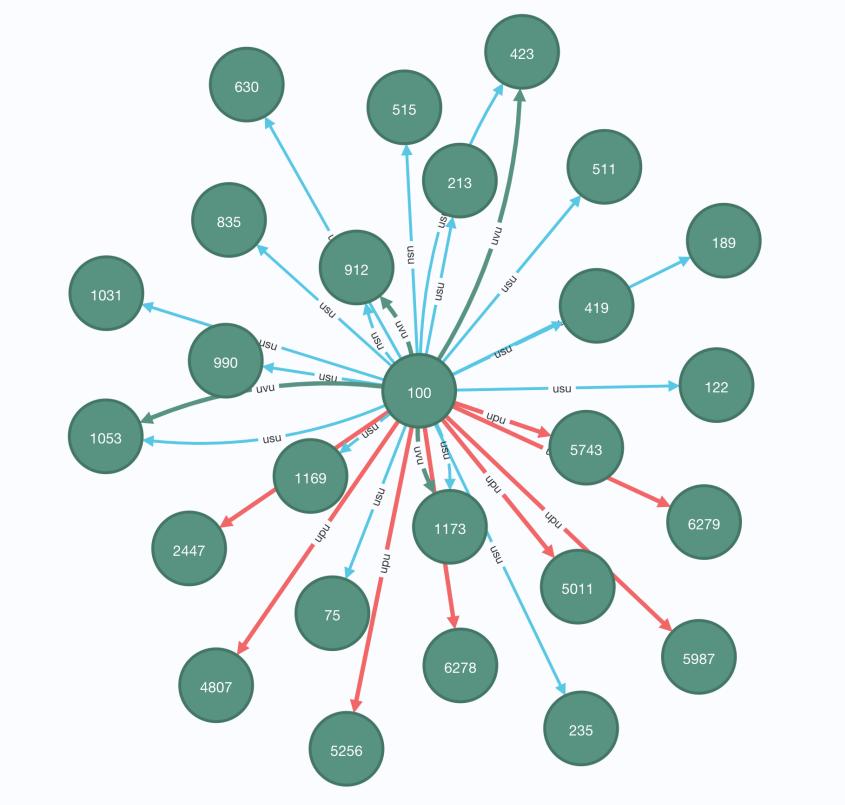
Nodes	%Fraud Nodes (Class=1)
11,944	9.5

Relation	Edges
U-P-U	175,608
U-S-V	3,566,479
U-V-U	1,036,737
All	4,398,392

Graph Visualization

Demo

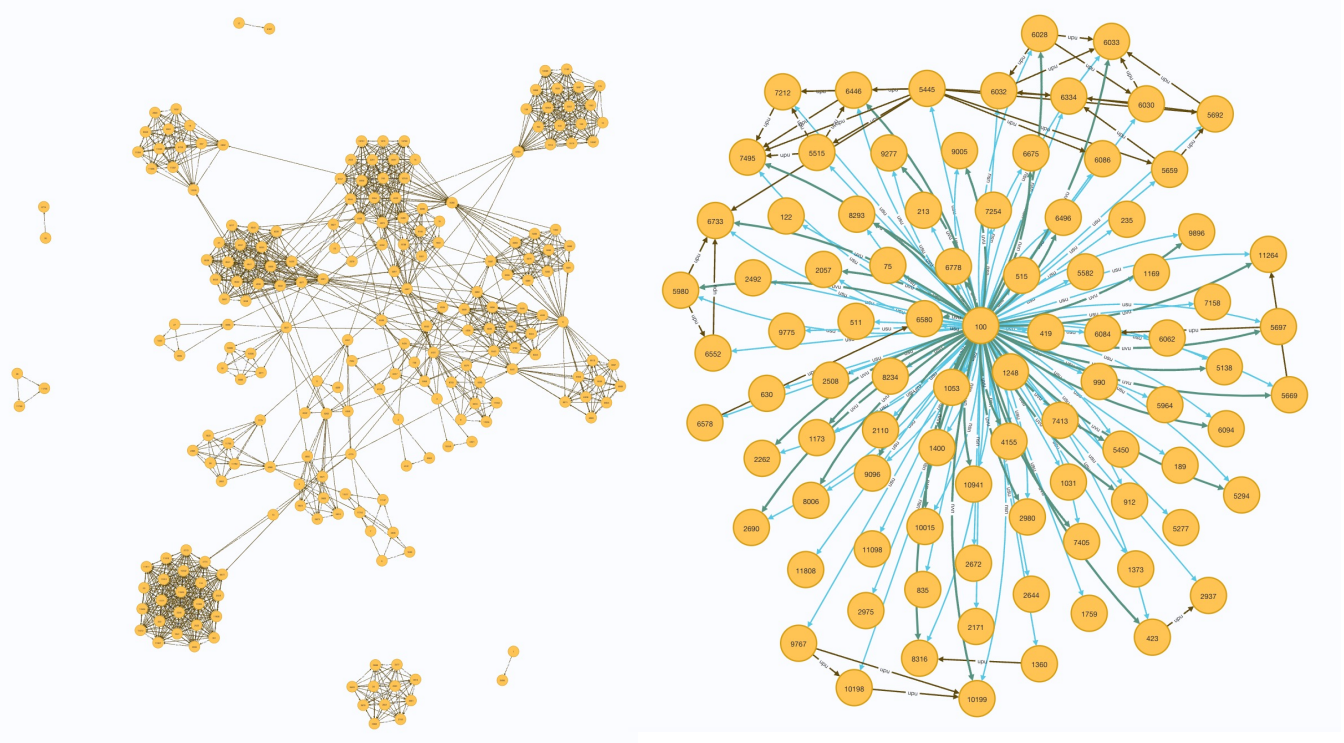
1) **U-P-U:** it connects users reviewing at least one same product
2) **U-S-U:** it connects users having at least one same star rating within one week
3) **U-V-U:** it connects users with top 5% mutual review text similarities (measured by TF-IDF) among all users



Graph Connections of Node ID100 (Neo4j)

Graph Visualization

Demo



Displaying 220 nodes, 200 relationships.

Models

Models

01

PC-GNN

Pick and Choose Graph Neural Network

02

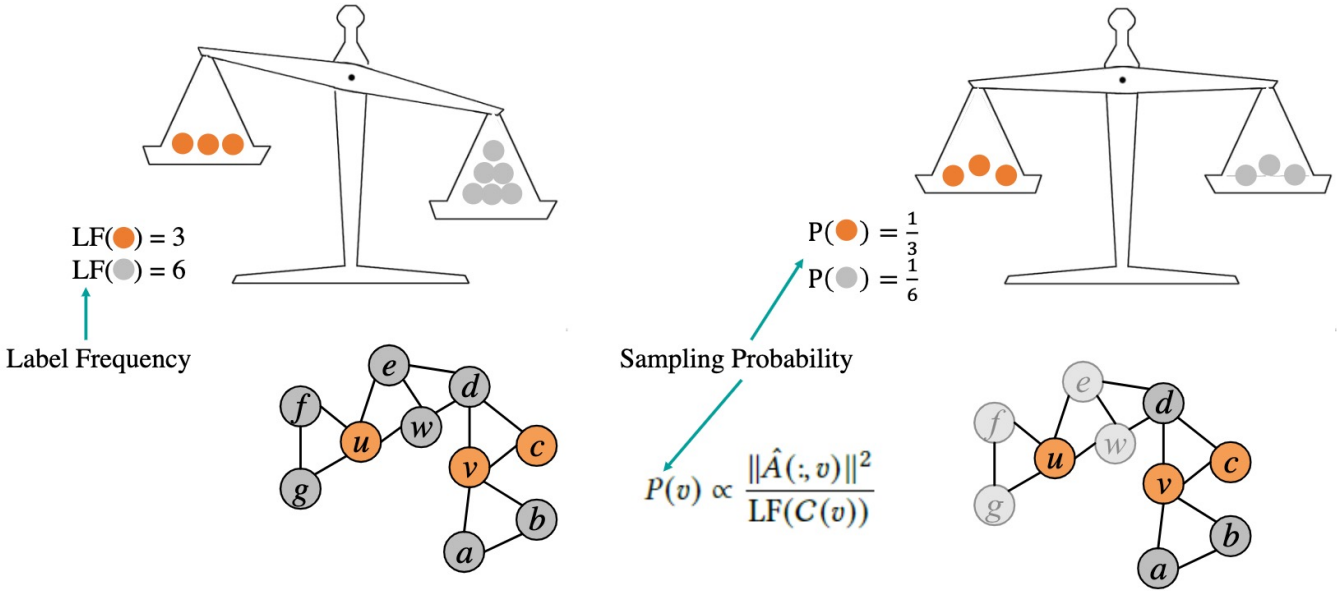
Rio-GNN

Reinforced Neighborhood Selection Guided Multi - relational Graph Neural Networks

PC-GNN

Methodology

Step 1: Pick nodes from whole graph

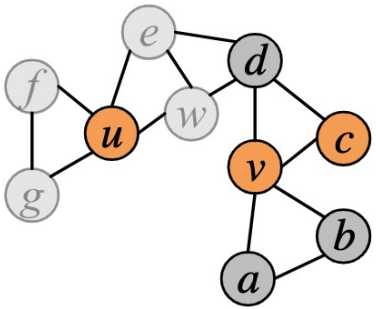


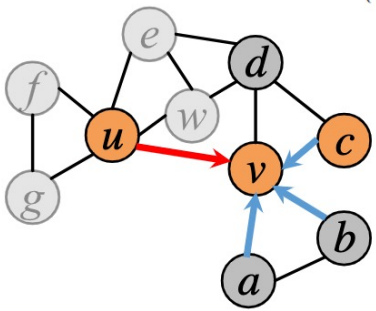
PC-GNN

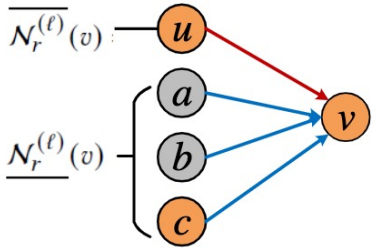
Methodology

Step 2: 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: $\mathcal{N}_r^{(\ell)}(v) = \underline{\mathcal{N}_r^{(\ell)}}(v)$

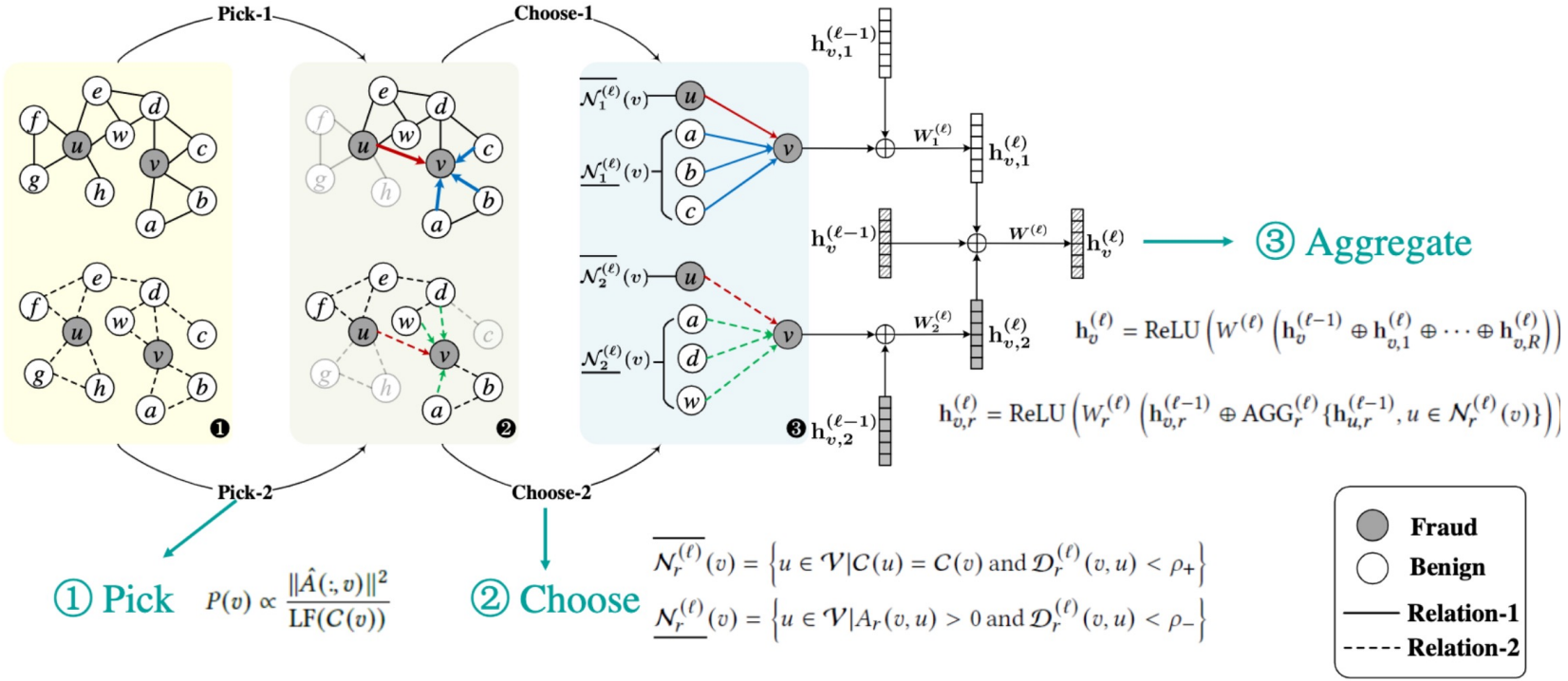
$$\mathcal{D}_r^{(\ell)}(v,u) = \left\| \mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{v,r}^{(\ell)}\right) - \mathrm{D}_r^{(\ell)}\left(\mathbf{h}_{u,r}^{(\ell)}\right) \right\|_1$$

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

PC-GNN

Methodology

Step 3: Aggregate



$$P(v) \propto \frac{\|\hat{A}(:,v)\|^2}{\mathrm{LF}(C(v))}$$

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

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

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

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

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

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

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

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

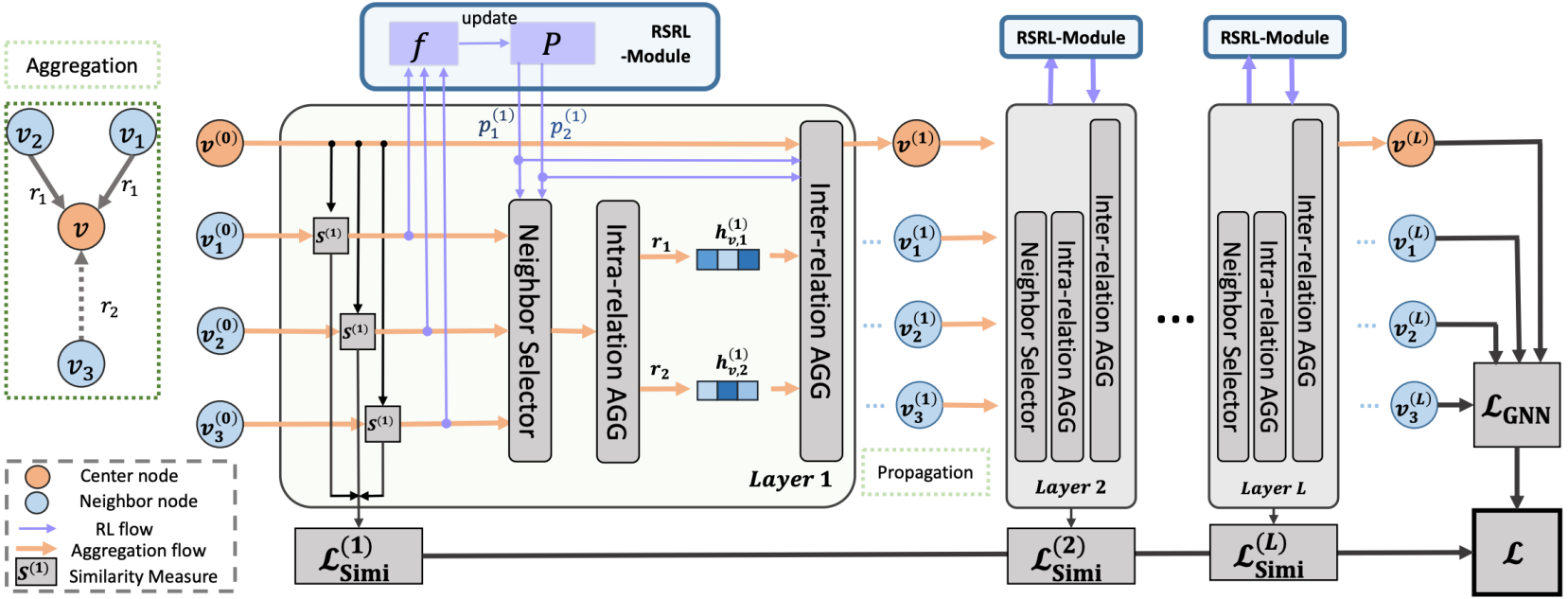
$$\mathcal{L}_{\text{gnn}} = -\sum_{v\in\mathcal{V}}\left[y_v \log p_v + (1-y_v)\log(1-p_v)\right]$$

- Overall loss function

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

Rio-GNN

Architecture



Rio-GNN

Loss Function

$$\mathcal{L}_{GNN} = \sum_{v \in \mathcal{V}} -log(y_v \cdot \sigma(MLP^{(l)}(\mathbf{z}_v))).$$

$$\mathcal{L}^{(l)}_{Simi} = \sum_{v \in \mathcal{V}} -log(y_v \cdot \sigma(FCN^{(l)}(\mathbf{h}^{(l)}_v))).$$

$$\mathcal{L}_{\text{RioGNN}} = \mathcal{L}_{GNN} + \lambda_l \sum_{l=1}^{L} \mathcal{L}^{(l)}_{Simi} + \lambda_* ||\Theta||_2,$$

Rio-GNN

Methodology

Step 1: Label-aware Neural Similarity Measure

Output: Similarity of 2 nodes

Formula: $\mathcal{D}^{(l)}(v, v') = ||\sigma(FCN^{(l)}\mathbf{h}_v^{(l-1)}) - \sigma(FCN^{(l)}\mathbf{h}_{v'}^{(l-1)})||_1.$

$$S^{(l)}(v, v') = 1 - \mathcal{D}^{(l)}(v, v').$$

Loss Function:

$$\mathcal{L}_{Simi}^{(l)} = \sum_{v \in \mathcal{V}} -log(y_v \cdot \sigma(FCN^{(l)}(\mathbf{h}_v^{(l)}))).$$

$$\mathcal{L}_{Simi} = \sum_{l=1}^{L} \mathcal{L}_{Simi}^{(l)}.$$

Rio-GNN

Methodology

Step 2: Similarity-aware Adaptive Neighbor Selector

Output: find top **p - sampling**, find optimal thresholds

- Find top p - sampling to filter dissimilar neighbors
- A filtering threshold $p\ l\ r \in [0, 1]$ for relation r at the l**-th layer** indicates the selection ratio from all neighbors
- Using **RSRL Framework** to find an optimal filtering thresholds

Rio-GNN

Methodology

Step 3: Relation-aware Weighted Neighbor Aggregator

Embedding under relation r:

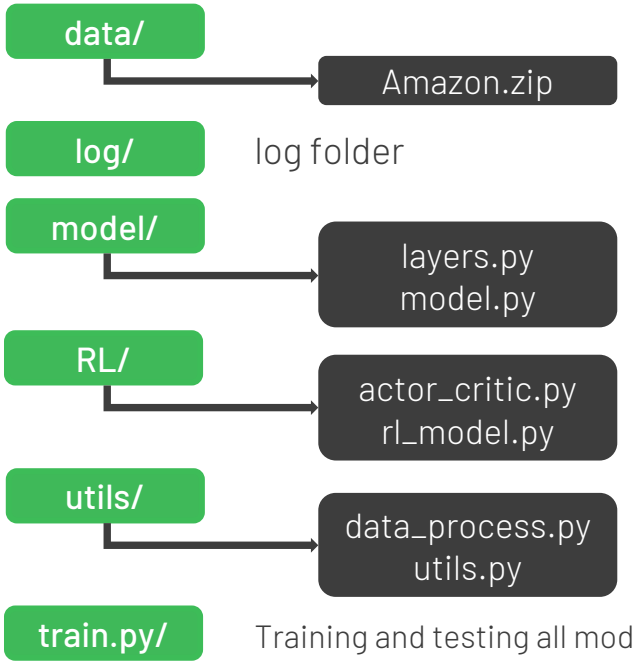
$$\mathbf{h}_{v,r}^{(l)} = ReLU(AGG_r^{(l)}(\{\oplus\, \mathbf{h}_{v'}^{(l-1)} : v' \in \mathcal{N}_r^l(v)\})),$$

Node embedding:

$$\mathbf{h}_v^{(l)} = ReLU(\mathbf{h}_v^{(l-1)} \oplus AGG^{(l)}(\{\oplus\, (p_r^{(l)} \cdot \mathbf{h}_{v,r}^{(l)})\}|_{r=1}^{R})),$$

Training Model

Repository Structure



Data of the dataset Amazon

RioGNN layers implementations
RioGNN model implementations

RL algorithm, Actor-Critic
RioGNN RL Forest implementations

Transfer sparse matrix to adjacency lists
Utility functions for data i/o and model evaluation

Training Model

Evaluation Metric

- We utilize ROC-AUC (AUC) and Recall to evaluate the overall performance of all classifiers. AUC is computed based on the relative ranking of prediction probabilities of all instances, which could eliminate the influence of imbalanced classes.
- The Recall is defined as:

$$Recall = \frac{TP}{TP + FN},$$

where TP is True Positive, FN is False Negative. The AUC is defined as:

$$AUC = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i)(y_i + y_{i+1}),$$

- In order to better measure the effectiveness, we add the F1 indicator to measure after the two indicators of AUC and Recall. F1 is defined as:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recal},$$

Training Model

PC-GNN

Two variants of PC-GNN to comprehensively compare and analyze the performances of its each component.

- **PC-GNN\$_P$** : removing label-balanced sampler and following the original label distribution in sub-graph sampling.
- **PC-GNN\$_C$**: removing neighborhood sampler and aggregating messages from all topological neighbors.

Training Model

Rio-GNN

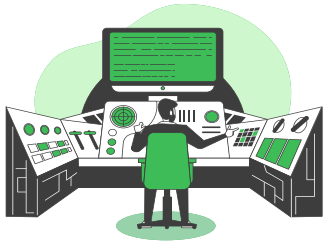
Three variants of Rio-GNN:

- **RIO-Att:** This variant uses single-layer similarity perception for neighbor selection, and uses the ActorCritic algorithm with a discrete strategy to recursively select the filter thresholds of different relations. But it chooses the method of Attention when aggregating neighbors between different relations.
- **RIO-Weight:** chooses the method of Weight when aggregating neighbors between different relations.
- **RIO-Mean:** chooses the method of Mean when aggregating neighbors between different relations.

Results

Performance Comparison

	AUC	F1-macro	F1-fraud	F1-benign
PC-GNN\$_P$	0.9469 ± 0.0018	**0.9158 ± 0.0024**	0.8463 ± 0.0045	0.9853 ± 0.0003
PC-GNN\$_C$	0.9529 ± 0.0035	0.8929 ± 0.0171	0.8070 ± 0.0295	0.9788 ± 0.0047
PC-GNN	**0.9586 ± 0.0014**	0.8956 ± 0.0077	0.8116 ± 0.0133	0.9795 ± 0.0021



Performance Comparison

	AUC	Recall
RIO-Att	93.97	83.78
RIO-Weight	**96.34**	88.46
RIO-Mean	94.57	89.47
RioGNN	96.16	88.66

RioGNN ✓

Conclusion

- This research studies RioGNN, a reinforced, recursive and flexible neighborhood selection guided multi-relational Graph Neural Network architecture, to learn more discriminative node embedding and respond to the explanation of the importance of different relations in spam review detection and disease diagnosis tasks, respectively.
- RioGNN designs a label-aware neural similarity neighbor measure and reinforced relation-aware neighbor selectors using reinforcement learning technology, respectively.
- Our work shows the promise in learning a reinforced neighborhood aggregation for GNNs, potentially opening new avenues for future research in boosting the performance of GNNs with adaptive neighborhood selection and analysing the importance of different relations in message passing.

Thanks for listening!