

# 图神经网络在金融领域的应用



中国科学院 信息工程研究所  
INSTITUTE OF INFORMATION ENGINEERING, CAS

汇报人：石逢钊 (2018级直博士生)  
指导老师：尚燕敏  
导师：曹亚男

# 目录

---

- 背景
- 欺诈检测
  - 问题介绍
  - 类别不平衡-PC-GNN: WWW21
  - 类被不平衡-AO-GNN: WWW22
  - 异配性-AMNet: IJCAI22
  - 异配性-GHRN: WWW23
- 股票预测-STHAN-SR: AAAI21

# 背景

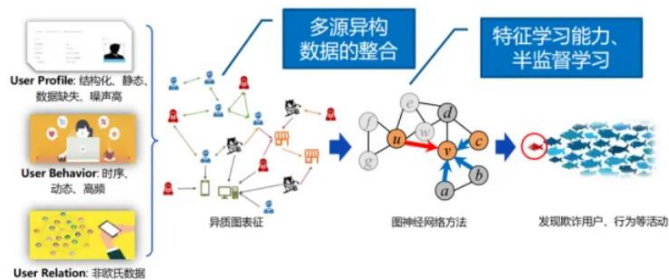
- 当前金融行业已经逐渐开始数字化进程，大量金融交易和金融事件发生在互联网中，同时大量的传统业务被迁移到线上。由此产生了比之前更多的金融问题。另外，随着机器学习的发展，相关技术也被应用于金融领域之中。
- 金融行业中不同个体之间产生了大量交互信息，这些交互信息将不同个体互相关联，形成了一个图结构，利用这些图结构能够很好的解决金融领域中的重要问题，同时图神经网络的大力发展也为解决这些问题带来了可能。

## 例子

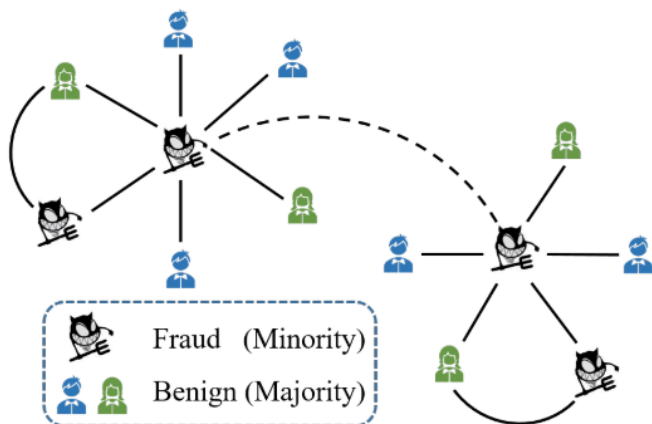


# 欺诈检测-主要问题

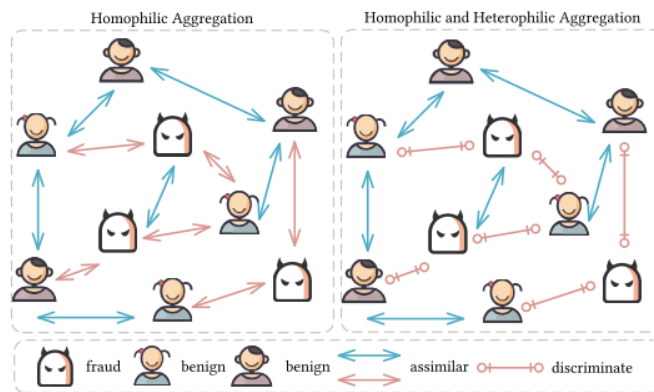
- 类别不平衡。
- 连边的异配性。
- 数据的多元异质性。



数据的多元异质性



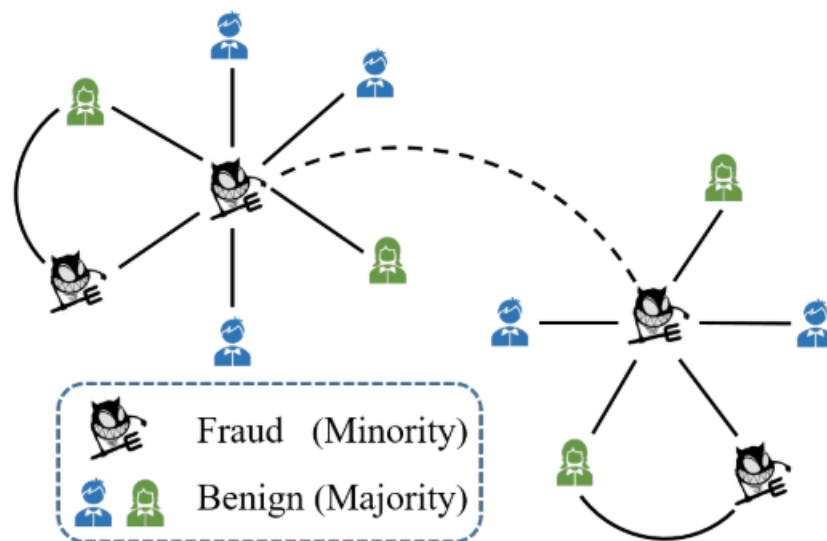
类别不平衡



连边的异配性

# 类别不平衡-PC-GNN: WWW21

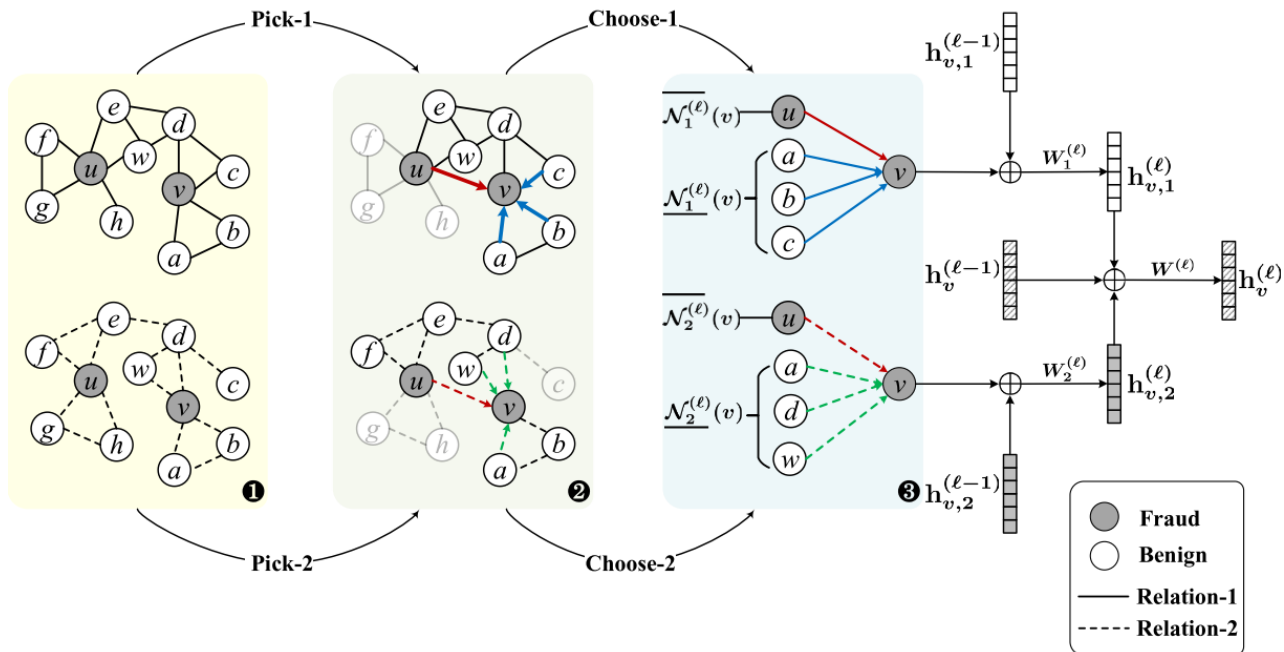
- 主要问题
  - 类别不平衡。
  - 欺诈者伪装造成的异配边影响聚合效果。



# 类别不平衡-PC-GNN: WWW21

## ● 模型设计

- 任务整体被抽象为在多关系图中进行二分类的任务。
- 包含三个部分，进行训练节点下采样的 Pick 部分，进行邻居选择的 Choose 部分，和最终实现卷积的表示学习部分



# 类别不平衡-PC-GNN: WWW21

## ● Pick

- 该部分的思想是利用下采样的方法，将训练集中的欺诈样例和正常样例通过采样的方法使得数目相等，其采样概率为：

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

## ● Choose

- 该部分通过下采样和上采样的方式来添加同配邻居，减少异配邻居。
- 设计邻居判别器对邻居进行判别：

$$\mathcal{D}_r^{(\ell)}(v, u) = \left\| D_r^{(\ell)}(\mathbf{h}_{v,r}^{(\ell)}) - D_r^{(\ell)}(\mathbf{h}_{u,r}^{(\ell)}) \right\|_1$$
$$\mathcal{L}_{\text{dist}} = - \sum_{\ell=1}^L \sum_{r=1}^R \sum_{v \in \mathcal{V}} \left[ y_v \log p_{v,r}^{(\ell)} + (1 - y_v) \log (1 - p_{v,r}^{(\ell)}) \right]$$
$$p_{v,r}^{(\ell)} = D_r^{(\ell)}(\mathbf{h}_{v,r}^{(\ell)})$$

- 然后按照阈值进行下采样和上采样（上采样仅仅训练时采用）：

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

- 聚合：采用单层GraphSage完成聚合操作

# 类别不平衡-PC-GNN: WWW21

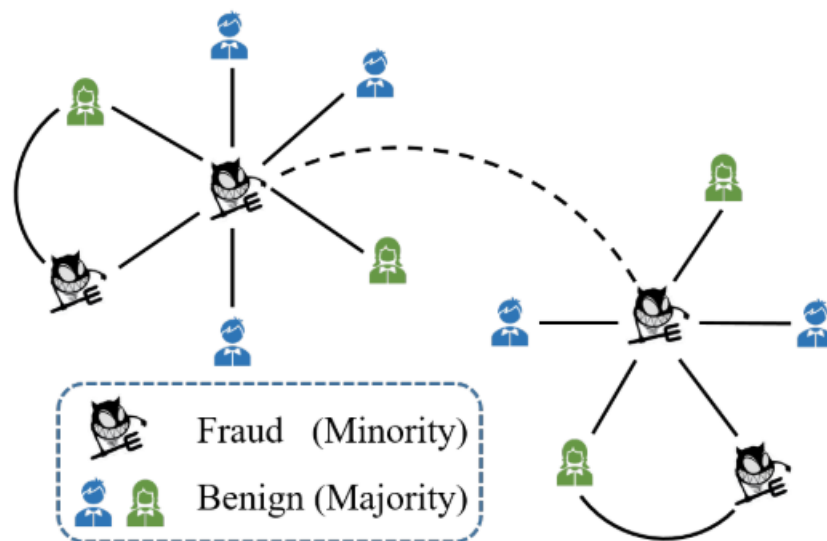
## ● 实验结果

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 <sub>\P</sub>	0.5136±0.0147	0.7844±0.0013	0.2336±0.0356	<b>0.9158±0.0024</b>	0.9469±0.0018	0.8782±0.0068
	PC-GNN <sub>\C</sub>	<b>0.6634±0.0058</b>	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	<b>0.7987±0.0014</b>	<b>0.7160±0.0130</b>	0.8956±0.0077	<b>0.9586±0.0014</b>	<b>0.9030±0.0044</b>



# 类别不平衡-AO-GNN: WWW22

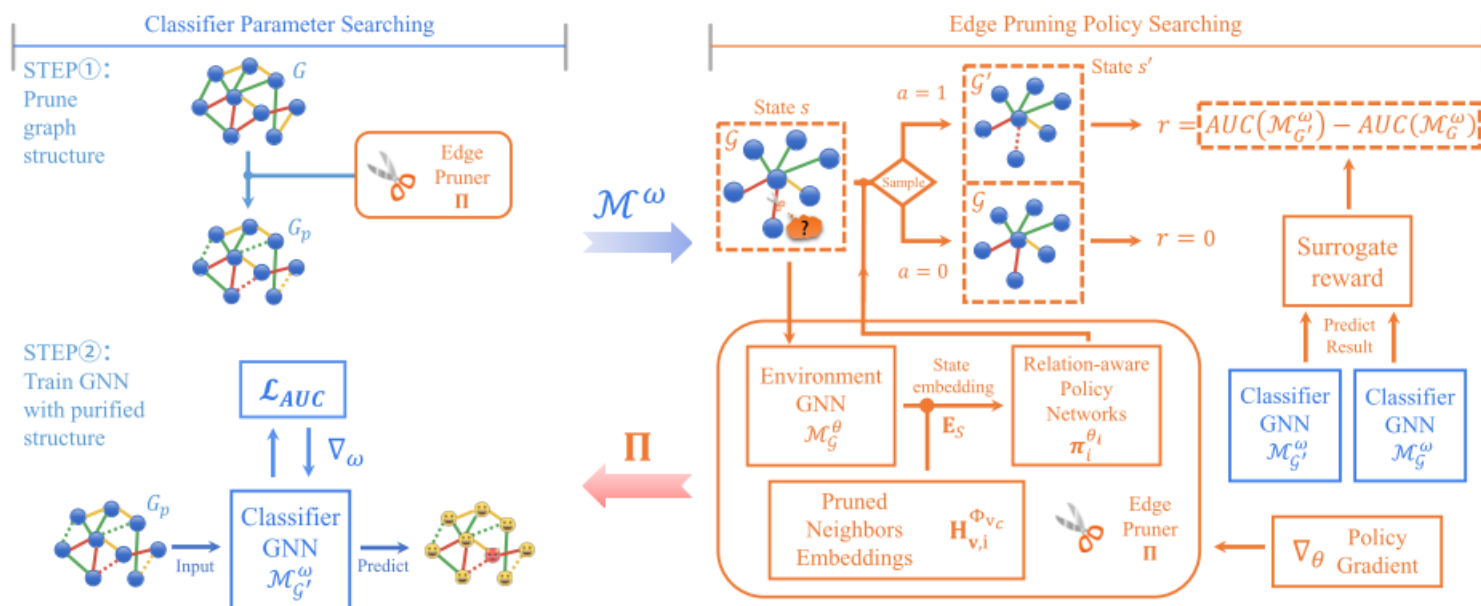
- 主要问题
  - 类别不平衡。
  - 欺诈者伪装造成的异配边影响聚合效果。



# 类别不平衡-AO-GNN: WWW22

## ● 模型结构

- 仍将问题作为多关系图上的二分类来做。
- 模型分为两部分：一部分是用于分类的GNN模型，另一部分则是边切除策略器。



# 类别不平衡-AO-GNN: WWW22

## ● 基本分类模型

- 本文采用了类GraphSage的模型作为基本模型进行节点嵌入的生成:

$$\mathbf{H}_{v,k}^{(l+1)} = \frac{1}{|N_k(\mathbf{v})|} \sum_{u \in N_k(\mathbf{v})} \mathbf{H}_u^{(l)},$$
$$\mathbf{H}_v^{(l+1)} = \sigma(W^{(l)}(\mathbf{H}_v^{(l)} \oplus \mathbf{H}_{v,1}^{(l+1)} \oplus \dots \oplus \mathbf{H}_{v,r}^{(l+1)})).$$

- 本文对不平衡训练的修改主要是修改了训练的loss, 不再使用交叉熵, 而是最大化AOC值:

$$\max_{\omega, \Pi} \mathbb{E}_{\mathbf{v}, \mathbf{v}'} (\mathbb{I}(\mathcal{M}_{\mathcal{G}}(\omega; \mathbf{v} | \Pi) \geq \mathcal{M}_{\mathcal{G}}(\omega; \mathbf{v}' | \Pi)) | y_{\mathbf{v}} = 1, y_{\mathbf{v}'} = 0)$$

$$\min_{\omega \in \mathbb{R}^d, \{a, b\} \in \mathbb{R}^2} \max_{\alpha \in \mathbb{R}} \mathbb{E}_{\mathbf{v}} [\mathcal{L}_{AUC}(\omega, a, b, \alpha, \mathbf{v} | \Pi | p)], \quad (9)$$

where  $p$  is the ratio of fraud nodes, and

$$\begin{aligned} \mathcal{L}_{AUC}(\omega, a, b, \alpha, \mathbf{v} | \Pi, p) &= \mathbb{I}(y = 1) [(1 - p)(\mathcal{M}_{\mathcal{G}}(\omega; \mathbf{v} | \Pi) - a)^2 \\ &\quad + 2(p - 1)(1 + \alpha)\mathcal{M}_{\mathcal{G}}(\omega; \mathbf{v} | \Pi)] \\ &\quad + \mathbb{I}(y = 0) [p(\mathcal{M}_{\mathcal{G}}(\omega; \mathbf{v} | \Pi) - b)^2 \\ &\quad + 2p(1 + \alpha)\mathcal{M}_{\mathcal{G}}(\omega; \mathbf{v} | \Pi)] + p(1 - p)\alpha^2. \end{aligned}$$

# 类别不平衡-AO-GNN: WWW22

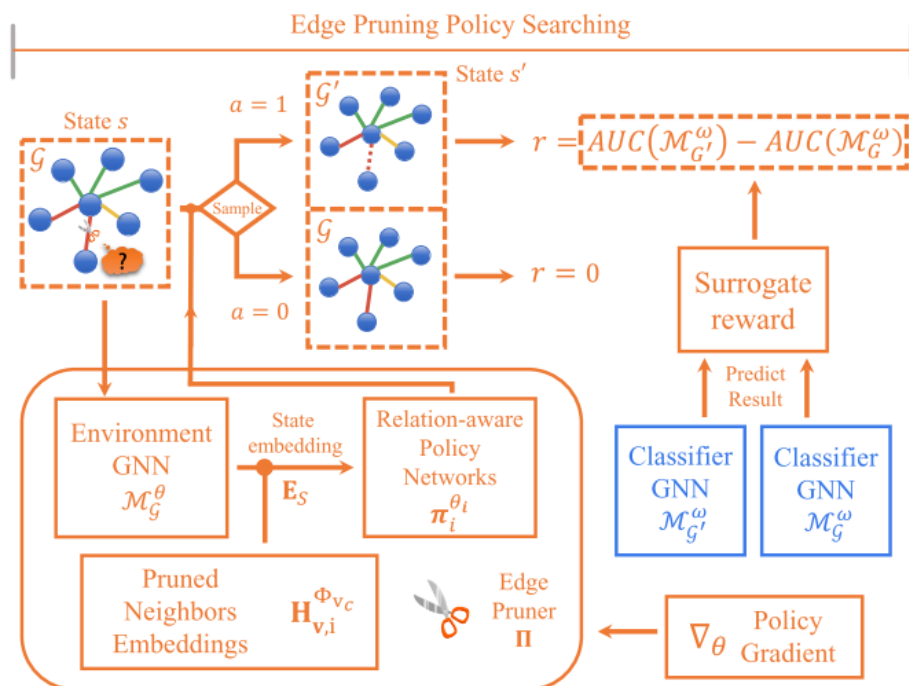
## ● 边切除策略器

- 该部分将连边的剪除当作是一个马尔可夫决策过程 (MDP)，然后每次针对某个节点的某条边来判定该条连边是否保留。显然这是个强化学习过程。
- 在整个过程中，模型使用一个训练一个策略选择器来判断是否保留某条边，采用已训练的GNN模型来给出每次判断的reward。

$$\mathbf{H}_{v,i}^{\Phi_{vc}} = \frac{1}{|\phi_i^{vc}|} \sum_{u \in |\phi_i^{vc}|} \mathcal{X}_u,$$

$$\mathbf{E}_s = \mathcal{X}_{v_p} \oplus \mathbf{H}_{v_c}^{(l)} \oplus \mathbf{H}_{v_p}^{(l)} \oplus \mathbf{H}_{v,1}^{\Phi_{vc}} \oplus \dots \oplus \mathbf{H}_{v,r}^{\Phi_{vc}}$$

$$\pi_i(a = 1 | s) = \text{Sigmoid}(\text{MLP}(\mathbf{E}_s))$$



# 类别不平衡-AO-GNN: WWW22

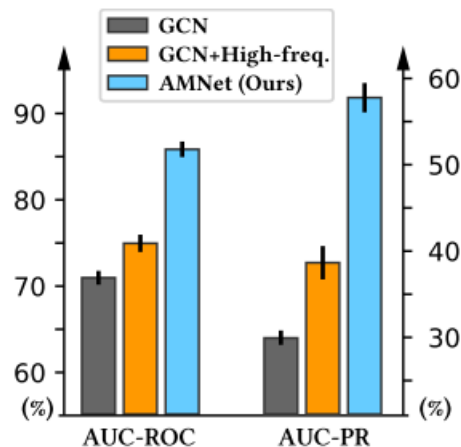
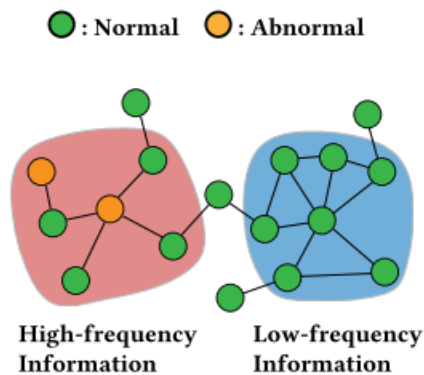
## ● 实验结果

Method	Dataset	YelpChi			Amazon			Books		
	Metric	AUC	F1-macro	GMean	AUC	F1-macro	GMean	AUC	F1-macro	GMean
Baselines	GCN	0.5983±0.0049	0.5620±0.0067	0.4365±0.0262	0.8369±0.0125	0.6408±0.0694	0.5718±0.1951	0.4538±0.1977	0.2374±0.2065	0.0000±0.0000
	GAT	0.5715±0.0029	0.4879±0.0230	0.1659±0.0789	0.8102±0.0179	0.6464±0.0387	0.6675±0.1345	0.4006±0.2023	0.2058±0.1623	0.0000±0.0000
	GraphSAGE	0.5439±0.0025	0.4405±0.1066	0.2589±0.1864	0.7589±0.0046	0.6416±0.0079	0.5949±0.0349	0.4761±0.1508	0.2464±0.2004	0.0000±0.0000
	DR-GCN	0.5921±0.0195	0.5523±0.0231	0.4038±0.0742	0.8295±0.0079	0.6488±0.0364	0.7963±0.0091	0.5131±0.1579	0.3048±0.2454	0.0000±0.0000
	GraphConsis	0.6983±0.0302	0.5870±0.0200	0.5857±0.0385	0.8741±0.0334	0.7512±0.0325	0.7677±0.0486	0.5647±0.1281	0.2912±0.1325	0.0000±0.0000
	CARE-GNN	0.7619±0.0292	0.6332±0.0094	0.6791±0.0359	0.9067±0.0112	<b>0.8990±0.0073</b>	0.8962±0.0018	0.6267±0.0462	0.4050±0.0996	0.4861±0.0811
	PC-GNN	0.8178±0.0014	0.6400±0.0230	0.7395±0.0130	0.9586±0.0014	0.8956±0.0077	0.9030±0.0044	0.6431±0.0189	0.4951±0.0037	0.5244±0.1012
Ablation	AO-GNN <sub>woP</sub>	0.8680±0.0020	<b>0.7182±0.0177</b>	0.7484±0.0125	0.9588±0.0008	0.8956±0.0026	0.8740±0.0137	0.6720±0.0111	0.4131±0.0102	0.4829±0.0519
	AO-GNN <sub>woC</sub>	0.8545±0.0177	0.7063±0.0129	0.7305±0.0241	0.9392±0.0166	0.8914±0.0041	0.8828±0.0267	0.5821±0.1397	0.2901±0.2102	0.3711±0.1919
	AO-GNN <sub>R-P</sub>	0.8302±0.0286	0.6936±0.0351	0.7192±0.0586	0.9197±0.0238	0.8827±0.0135	0.8602±0.0164	0.5604±0.1733	0.2845±0.2329	0.3068±0.1240
Ours	AO-GNN	<b>0.8805±0.0008</b>	0.7042±0.0051	<b>0.8134±0.0232</b>	<b>0.9640±0.0020</b>	0.8921±0.0045	<b>0.9096±0.0105</b>	<b>0.7174±0.0158</b>	<b>0.5503±0.0141</b>	<b>0.6127±0.0252</b>

# 异配连边-AMNet: IJCAI 22

## ● 主要问题

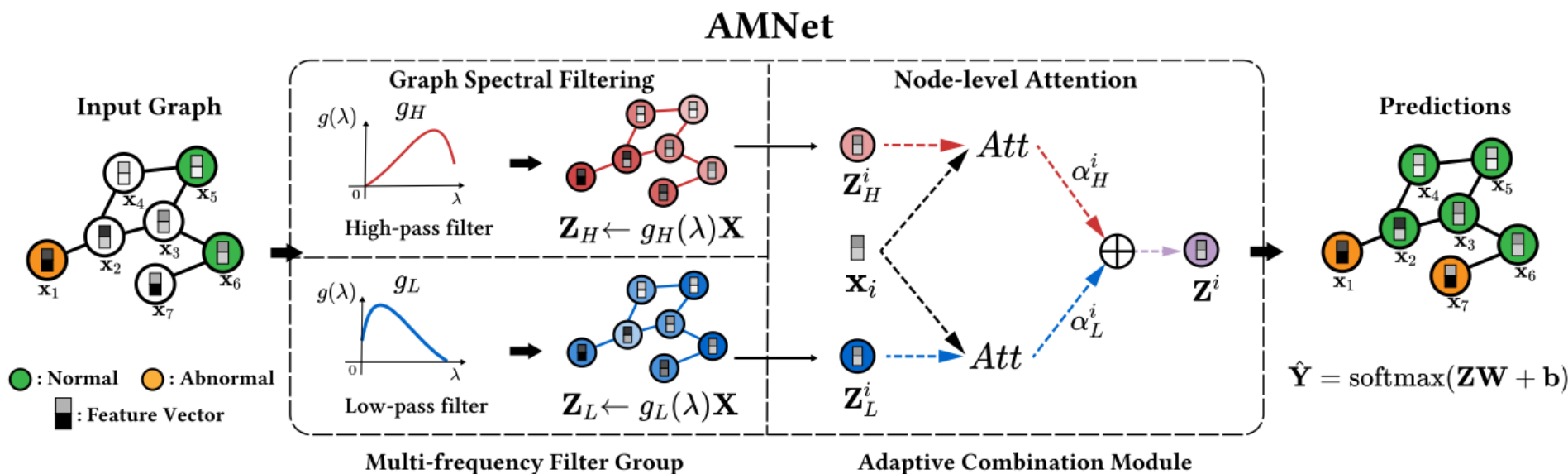
- 欺诈用户与正常用户在图上所表现的特征频率是不同的。
- 能否通过自适应地融合不同频率的特征来识别欺诈者



# 异配连边-AMNet: IJCAI 22

## ● 模型设计

- 模型是在一张图上做的二分类任务。
- 模型包括两个部分：一部分是频率学习器，一部分是频率信号融合。



# 异配连边-AMNet: IJCAI 22

- 频率学习器

- 该部分利用Bernstein多项式来分别学习欺诈节点和正常节点的频谱信号:

$$Z_k = \sum_{m=0}^M \theta_m \frac{1}{2^m} \binom{M}{m} (2-L)^{M-m} L^m f(X)$$

- 为了使得模型学到对应的频率, 添加loss:

$$\mathcal{L}_a = \sum_i \max(0, r_i (\alpha_L^i - \alpha_H^i) + \zeta)$$

- 频率特征融合

$$\omega_k^i = \mathbf{q}^T \cdot \tanh(\mathbf{W}^Z \mathbf{z}_k^i + \mathbf{W}^X \mathbf{x}_i)$$

$$\alpha_k^i = \text{softmax}(\omega_k^i) = \frac{\exp(\omega_k^i)}{\sum_k \exp(\omega_k^i)}$$

$$\mathbf{Z} = \sum_k \alpha_k \mathbf{Z}_k$$



# 异配连边-AMNet: IJCAI 22

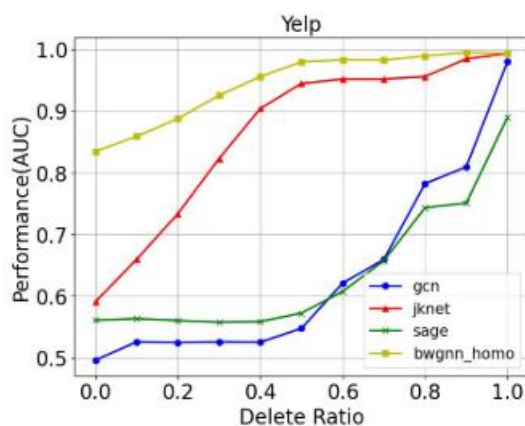
## ● 实验结果

Dataset \ Methods	Yelp		Elliptic		FinV		TeleCom	
	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC	AUC-PR
Graph Neural Networks								
GCN	$70.97 \pm 0.8$	$29.93 \pm 0.6$	$84.57 \pm 0.4$	$33.17 \pm 0.3$	$64.64 \pm 1.1$	$9.04 \pm 0.3$	$76.69 \pm 1.2$	$59.85 \pm 1.2$
GAT	$74.68 \pm 1.3$	$35.44 \pm 1.1$	$86.03 \pm 1.5$	$56.81 \pm 0.9$	$65.97 \pm 1.5$	$9.44 \pm 0.2$	$79.15 \pm 1.8$	$64.43 \pm 0.5$
GraphSAGE	$73.65 \pm 0.8$	$36.11 \pm 0.7$	$85.28 \pm 2.1$	$55.29 \pm 1.3$	$72.13 \pm 1.9$	$16.54 \pm 0.9$	$76.02 \pm 1.2$	$64.07 \pm 0.7$
GIN	$68.50 \pm 1.3$	$31.22 \pm 1.3$	$85.11 \pm 1.3$	$37.34 \pm 1.3$	$67.44 \pm 1.3$	$20.02 \pm 1.3$	$76.51 \pm 1.3$	$59.48 \pm 1.3$
GNN-based Graph Anomaly Detection Models								
DOMINANT	$49.32 \pm 0.8$	$15.58 \pm 0.3$	$16.21 \pm 0.3$	$5.48 \pm 0.1$	$64.59 \pm 1.1$	$8.28 \pm 0.3$	$55.43 \pm 0.7$	$15.68 \pm 0.3$
GeniePath	$75.89 \pm 1.8$	$35.86 \pm 0.5$	$83.14 \pm 1.3$	$44.37 \pm 0.8$	$72.27 \pm 1.2$	$18.43 \pm 0.7$	$83.73 \pm 0.7$	$64.25 \pm 0.3$
GraphConsis	$70.40 \pm 1.3$	$27.02 \pm 0.8$	$86.14 \pm 1.1$	$62.04 \pm 1.2$	$72.82 \pm 1.2$	$27.07 \pm 1.0$	$77.91 \pm 1.5$	$61.82 \pm 0.5$
CARE-GNN	$78.41 \pm 1.5$	$38.90 \pm 1.1$	$85.84 \pm 1.2$	$49.81 \pm 1.2$	$70.31 \pm 1.8$	$23.61 \pm 0.3$	$81.02 \pm 0.7$	$68.06 \pm 1.6$
AMNet	<b><math>85.85 \pm 1.1</math></b>	<b><math>57.77 \pm 0.9</math></b>	<b><math>88.52 \pm 1.0</math></b>	<b><math>74.62 \pm 1.4</math></b>	<b><math>78.38 \pm 1.8</math></b>	<b><math>29.31 \pm 0.8</math></b>	<b><math>87.62 \pm 1.3</math></b>	<b><math>75.18 \pm 0.9</math></b>

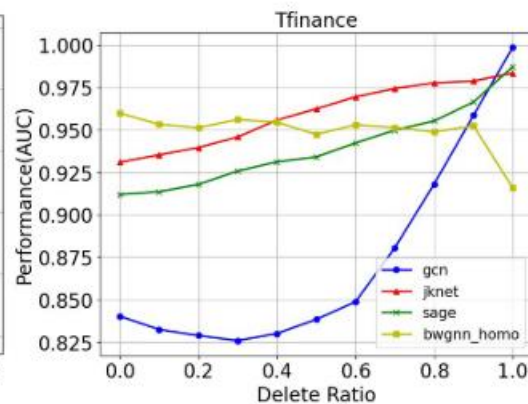
# 异配连边-GHRN: WWW23

## ● 主要问题

- 本文期望能够消除类间的连边从而实现更好地欺诈检测



(a) YelpChi



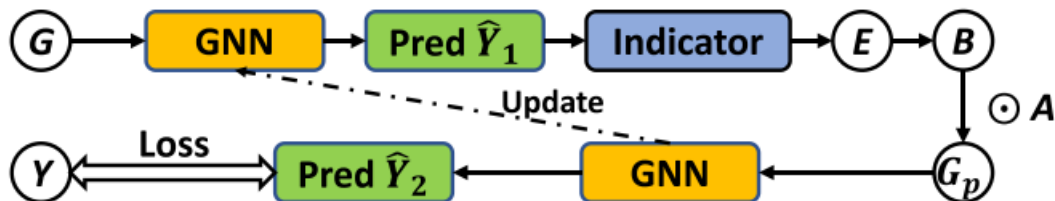
(b) T-Finance

# 异配连边-GHRN: WWW23

## ● 模型设计

- 本文核心即是设计一个删除类间连边的方法，尽可能减少异配边，从而在生成的新图上面重新训练模型，从而实现更好地分类。
- 其删除连边的基准便是：正常节点的类型间连边往往很少，而欺诈节点的类型间连边往往很多，依据与邻居的差异性既可以判别是否为类间连边

	normal	anomaly	heterophily of the graph
YelpChi	0.1317	0.8144	0.2268
Amazon	0.0234	0.9254	0.0456
T-Finance	0.0150	0.5280	0.0292
T-Social	0.2366	0.9161	0.3761



# 异配连边-GHRN: WWW23

- 邻居差异性计算公式

- Post-aggregation (PA) score:  $S = \hat{L}Y$  即:  $S_i = SIGN * [\frac{d_i}{d_i + 1}hetero(i), -\frac{d_i}{d_i + 1}hetero(i)]$

- 由上式可得:

$$S_{v \in \mathcal{V}_a} \cdot S_{v \in \mathcal{V}_a} > S_{v \in \mathcal{V}_n} \cdot S_{v \in \mathcal{V}_n} > 0 > S_{v \in \mathcal{V}_a} \cdot S_{v \in \mathcal{V}_n}$$

- 由于测试集合是不知道标签的, 因此用伪标签代替:

$$\hat{Y}_v = \begin{cases} [1 - \Delta_v, \Delta_v], y_v = 0 \\ [\Delta_v, 1 - \Delta_v], y_v = 1 \end{cases}$$

- 最后可得到计算的整体方法:  $E = L\hat{Y}\hat{Y}^T L^T$

- 类间边删除

- 在得到E之后按照E中的值删除其最小的K个得到新的图。

- 然后让模型在新图中重新运行, 得到最终结果。

# 异配连边-GHRN: WWW23

## ● 实验结果

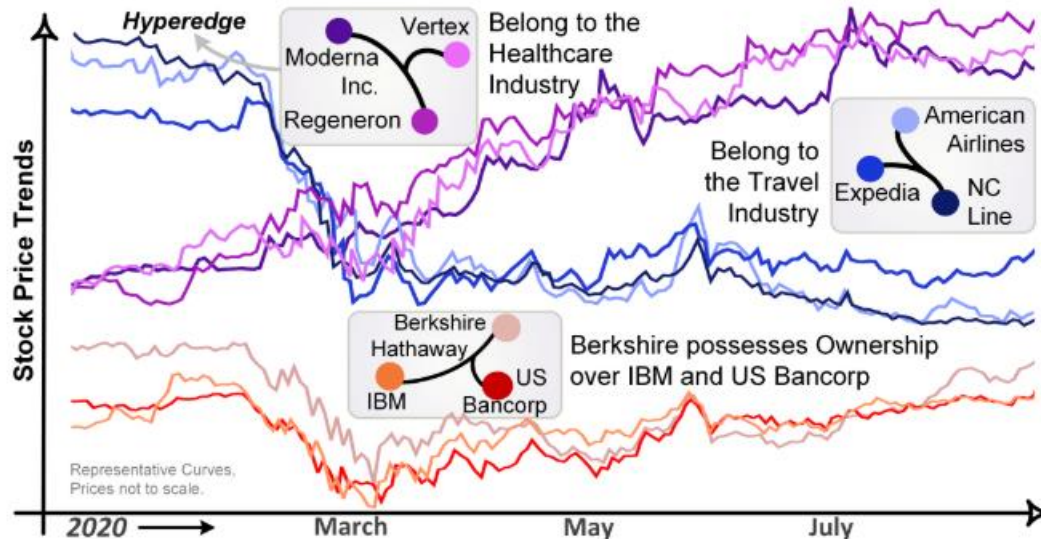
Method	Dataset	YelpChi		Amazon		T-finance		T-Social	
	Metric	F1-Macro	AUC	F1-Macro	AUC	F1-Macro	AUC	F1-Macro	AUC
Homophilous GNNs	MLP	0.4614	0.7366	<u>0.9010</u>	0.9082	0.4883	0.8609	0.4406	0.4923
	GCN [17]	0.5157	0.5413	0.5098	0.5083	0.5254	0.8203	<u>0.6550</u>	0.7012
	GAT [36]	0.4614	0.5459	0.5675	0.7731	0.8816	0.9388	0.4921	0.4923
	JKNet [43]	0.5805	0.7736	0.8270	0.8970	0.8971	0.9554	0.4923	<u>0.7226</u>
	JK-GHRN (Ours)	<u>0.6145</u>	<u>0.7765</u>	0.8756	<u>0.9206</u>	<u>0.9015</u>	<u>0.9559</u>	0.4923	0.7016
GAD Models	CARE-GNN [12]	0.5015	0.7300	0.6313	0.8832	0.6115	0.8731	0.4868	0.7939
	PC-GNN [23]	0.6925	0.8118	0.8367	<u>0.9555</u>	0.5322	0.9182	0.4536	0.8917
	PC-GHRN (Ours)	<u>0.7082</u>	<u>0.8230</u>	<u>0.8855</u>	0.9519	<u>0.6177</u>	<u>0.9238</u>	0.6218	0.9035
Heterophilous GNNs	H2GCN [49]	0.6575	0.8406	0.9213	0.9693	0.8824	0.9553	-	-
	MixHop [1]	0.6534	0.8796	0.8093	0.9723	0.4880	0.9569	0.6471	0.9597
	GPRGNN [8]	0.6423	0.8355	0.8059	0.9358	0.8507	<b>0.9642</b>	0.5976	<b>0.9722</b>
	BWGNN(Homo) [35]	0.6935	0.8255	0.9194	0.9395	0.8899	0.9599	<b>0.9145</b>	0.9630
	BHomo-GHRN (Ours)	<u>0.7532</u>	<u>0.8631</u>	<u>0.9203</u>	<u>0.9609</u>	<b>0.8975</b>	0.9609	0.9118	0.9637
	BWGNN(Hetero) [35]	0.7568	0.8967	0.9204	0.9706	-	-	-	-
	BHetero-GHRN (Ours)	<b>0.7789</b>	<b>0.9073</b>	<b>0.9282</b>	<b>0.9728</b>	-	-	-	-

# 股票预测-STHAN-SR:AAA121

## ● 主要问题

- 现有的股票预测方法未能将收益作为直接的评价标准
- 现有方法未能考虑股票之间的关联性对股票整体的影响。

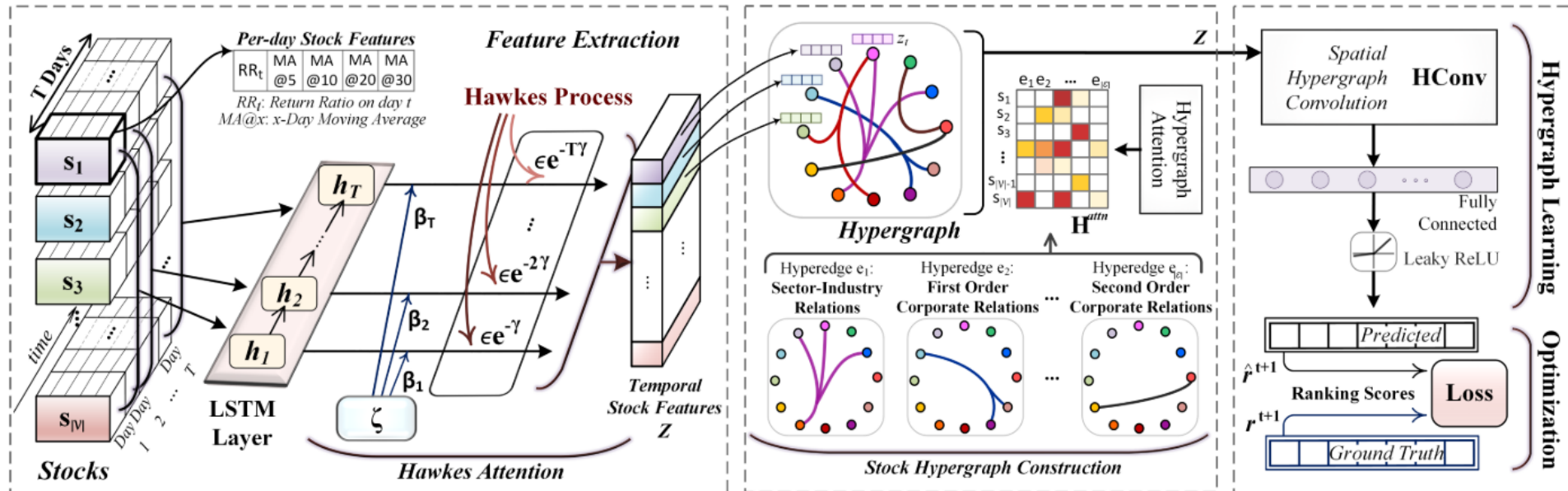
Stocks:	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	Performance	Profit(\$) <sup>↑</sup> Top-1 Stock buy-hold-sell
Returns <b>Ground Truth</b>	+50 <sup>↑</sup>	+30 <sup>↑</sup>	-10 <sup>↓</sup>	-35 <sup>↓</sup>		
Predicted return <i>(Regression Methods)</i> <b>R<sub>1</sub></b>	<b>60</b>	10	10	-45	250	<b>50</b> <sup>↑</sup>
<b>R<sub>2</sub></b>	25	<b>35</b>	-10	-40	<b>168</b> <sup>↓</sup>	30
Probability of change in return <i>(Classification Methods)</i> <b>C<sub>1</sub></b>	<sup>↓</sup> 0.6	<sup>↑</sup> 0.85	<sup>↓</sup> 0.55	<sup>↓</sup> 0.8	<b>75%</b> <sup>↑</sup>	30
<b>C<sub>2</sub></b>	<sup>↑</sup> 0.8	<sup>↓</sup> 0.7	<sup>↑</sup> 0.65	<sup>↓</sup> 0.75	50%	<b>50</b> <sup>↑</sup>



# 股票预测-STHAN-SR: AAAI 21

## ● 模型架构

- 模型整体分为两部分：一是特征提取部分，二是超图卷积生成节点表示部分。
- 模型按照每次交易的收益大小来最终输出每支股票的rank排名，从而得出推荐的股票。



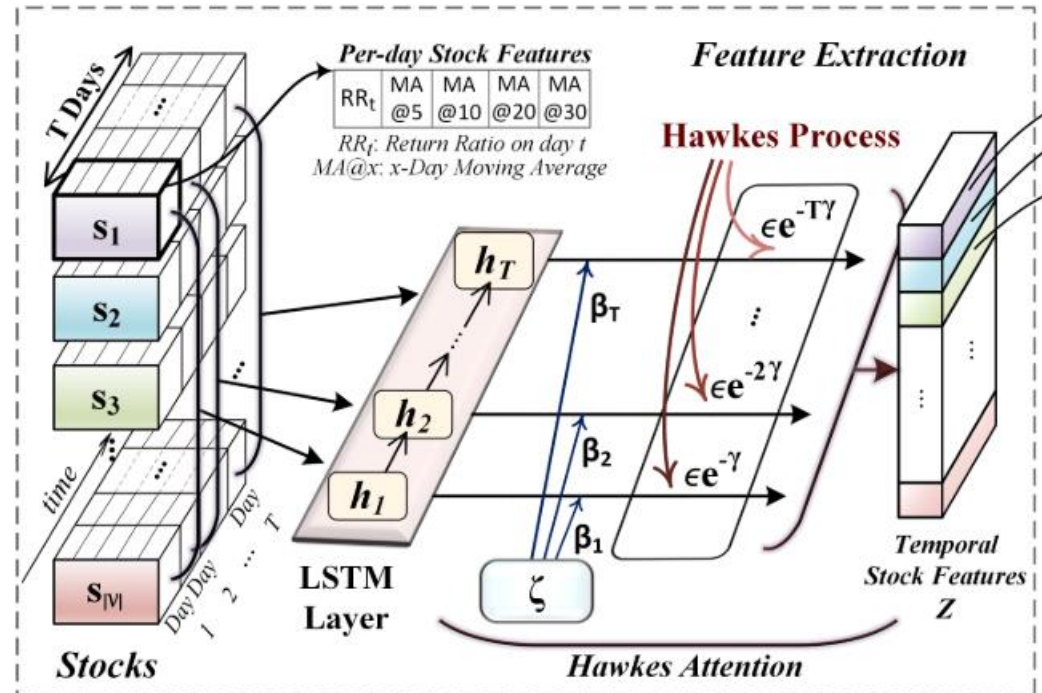
# 股票预测-STHAN-SR: AAAI 21

## ● 特征提取部分

- 首先模型提取了T次交易的特征作为初始特征，每个特征有五维，分别是股票在该时刻1天，5天，10天，20天，1个月的回报率。
- 然后使用LSTM进行特征提取。
- 之后采用时序attention+Hawkes Attention的方式将T次的交易特征融合为1.

$$\zeta(\bar{h}_t) = \sum_{\tau} \lambda_{\tau}, \lambda_{\tau} = \beta_{\tau} h_{\tau}, \beta_{\tau} = \frac{\exp(h_{\tau}^T W \bar{h}_t)}{\sum_{\tau} \exp(h_{\tau}^T W \bar{h}_t)}$$

$$z_t = \sum_{\tau=0, \Delta t_{\tau} \geq 0} (\lambda_{\tau} + \epsilon \max(\lambda_{\tau}, 0) e^{-\gamma \Delta t_{\tau}})$$

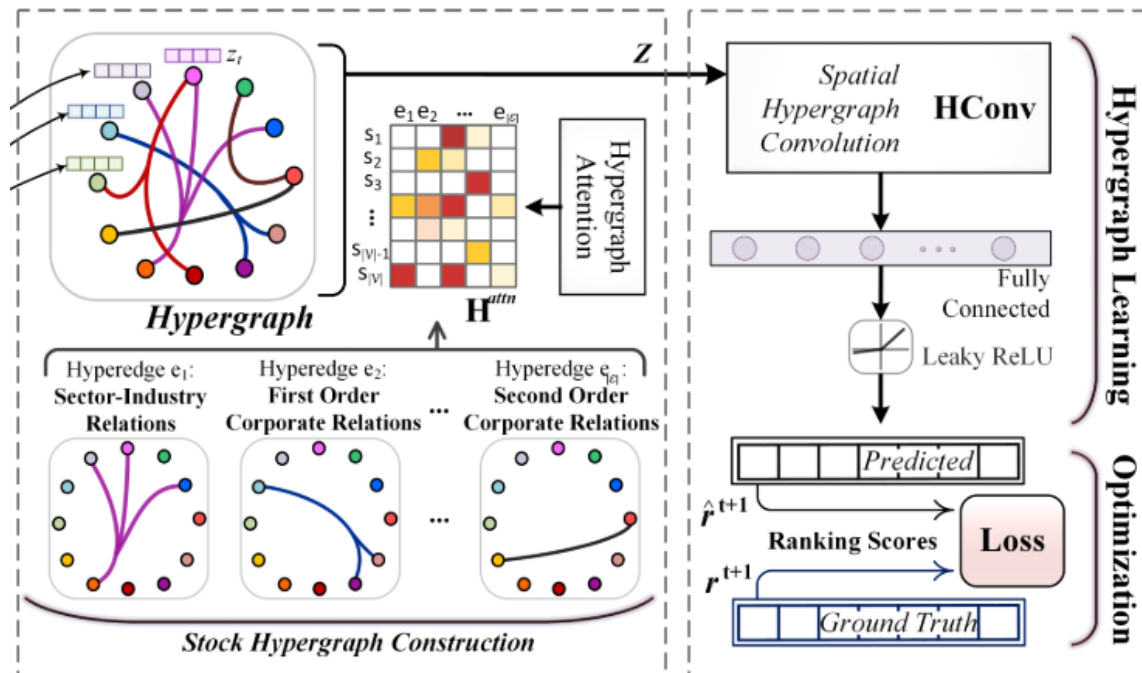




# 股票预测-STHAN-SR: AAAI 21

## ● 超图卷积

- 在这里模型设计了两类超图，一是如果两只股票属于统一企业就构成一条超边，由此构造第一个超图，二是通过Wiki中的数据来构造超图，即Wiki中显示两只股票有1-hop或2-hop关系则分别构造超图，这样就构造了三张超图。
- 然后分别在三张超图上做超图卷积生成表示，最后将其拼接融合，与传统的超图卷积不同的是它在进行节点特征生成时采用了attention机制。



# 股票预测-STHAN-SR: AAAI 21

## ● 实验结果

Model	Description	NASDAQ		NYSE		TSE	
		SR↑	IRR↑	SR↑	IRR↑	SR↑	IRR↑
CLF	ARIMA (Wang and Leu 1996)	0.55	0.10	0.33	0.10	0.47	0.13
	A-LSTM (Feng et al. 2019a)	0.97	0.23	0.81	0.14	1.10	0.43
	HGCluster (Luo et al. 2014)	0.06	0.10	0.10	0.11	0.20	0.10
	GCN (Li et al. 2020)	0.75	0.13	0.70	0.10	0.90	0.28
	HATS (Kim et al. 2019)	0.80	0.15	0.73	0.12	0.96	0.31
REG	SFM (Zhang et al. 2017)	0.16	0.09	0.19	0.11	0.08	0.07
	LSTM (Bao et al. 2017)	0.48	0.13	0.13	0.09	0.63	0.20
RL	DQN (Carta et al. 2020)	0.93	0.20	0.72	0.12	1.08	0.31
	iRDPG (Liu et al. 2020)	1.32	0.28	0.85	0.18	1.10 <sup>†</sup>	0.55 <sup>†</sup>
RAN	LSTM (Bao et al. 2017)	0.95	0.22	0.79	0.12	0.73	0.21
	GCN (Kipf et al. 2016)	0.46	0.13	0.72	0.16	0.81	0.27
	RSR-E (Feng et al. 2019b)	1.12	0.26	0.88	0.20	1.07	0.50
	RSR-I (Feng et al. 2019b)	1.34 <sup>*</sup>	0.39 <sup>*</sup>	0.95 <sup>*</sup>	0.21 <sup>*</sup>	1.08	0.53
	STHAN-SR (Ours)	1.42 <sup>*†</sup>	0.44 <sup>*†</sup>	1.12 <sup>*†</sup>	0.33 <sup>*†</sup>	1.19 <sup>*†</sup>	0.62 <sup>*†</sup>

# 后续讨论班介绍

---

- 第一周：图神经网络在金融领域的应用-石逢钊
- 第二周：图神经网络赋能的知识图谱研究与应用-刘 瑜
- 第三周：图网络在生物化学领域的应用-周玉晨
- 第四周：图网络在社会网络中的应用-宋传承
- 第五周：图神经网络在推荐系统中的应用-吴咏萱