F²GNN: An Adaptive Filter with Feature Segmentation for Graph-Based Fraud Detection

Guanghui Hu[1,2], Yang Liu[*2], Qing He[1,2] and Xiang Ao[*1,2,3]

1. Henan Institute of Advanced Technology, Zhengzhou University, Zhengzhou 450002, P.R. China
2. Key Laboratory of AI Safety & Security, Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
3. CASMINO Ltd., Suzhou 215000, China

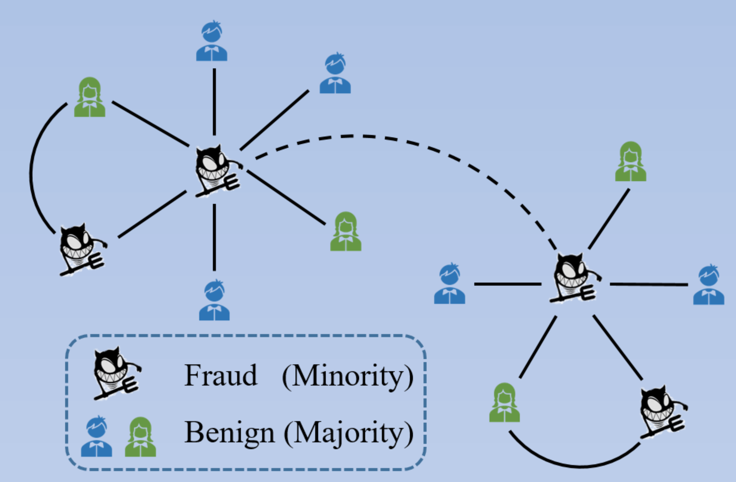
https://github.com/hugh925/F2GNN

Contribution

- We propose F²GNN to segment user features to mine hidden fraudulent information for graph-based fraud detection.
- We utilize **adaptive graph filters** on each segmented feature to model fraudulent behaviors and effectively address the class imbalance issue.
- Experiments on two real-world datasets for graph-based fraud detection validate the effectiveness of F²GNN.

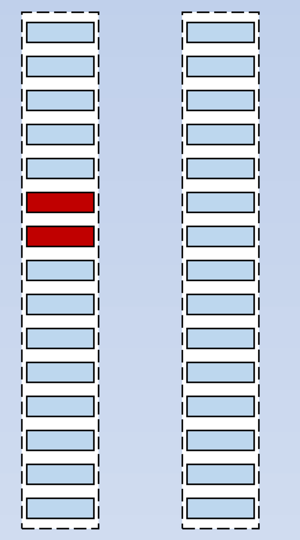
Problem Statement

- **Graph-based fraud detection** is a semi-supervised binary node classification problem on a graph.
- Each node v represents the target entity, which has a label $y_v \in \{0, 1\}$. The label 0 represents benign and 1 represents fraud.



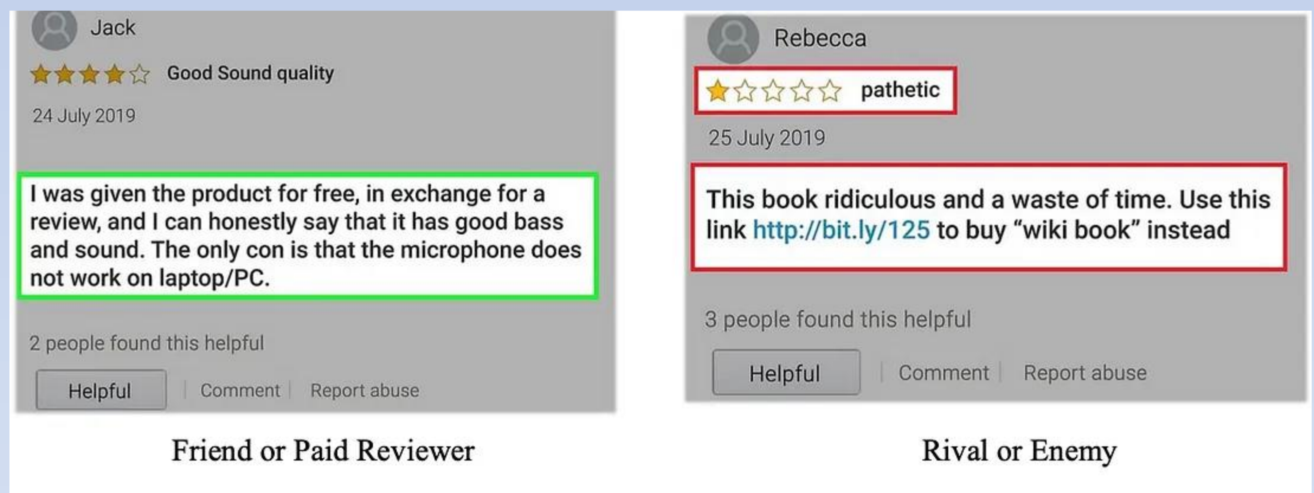
Motivation

- Fraudsters actively camouflage to evade detection, which may result in behaviors representing fraudulent characteristics **occupying only a small dimension** of the overall features.
- The number of fraudsters and the feature dimensions representing fraudulent behavior are both scarce, and the smoothing properties of low-pass filters may not be suitable for handling this task.



Datasets

Product Review Datasets: YelpChi and Amazon



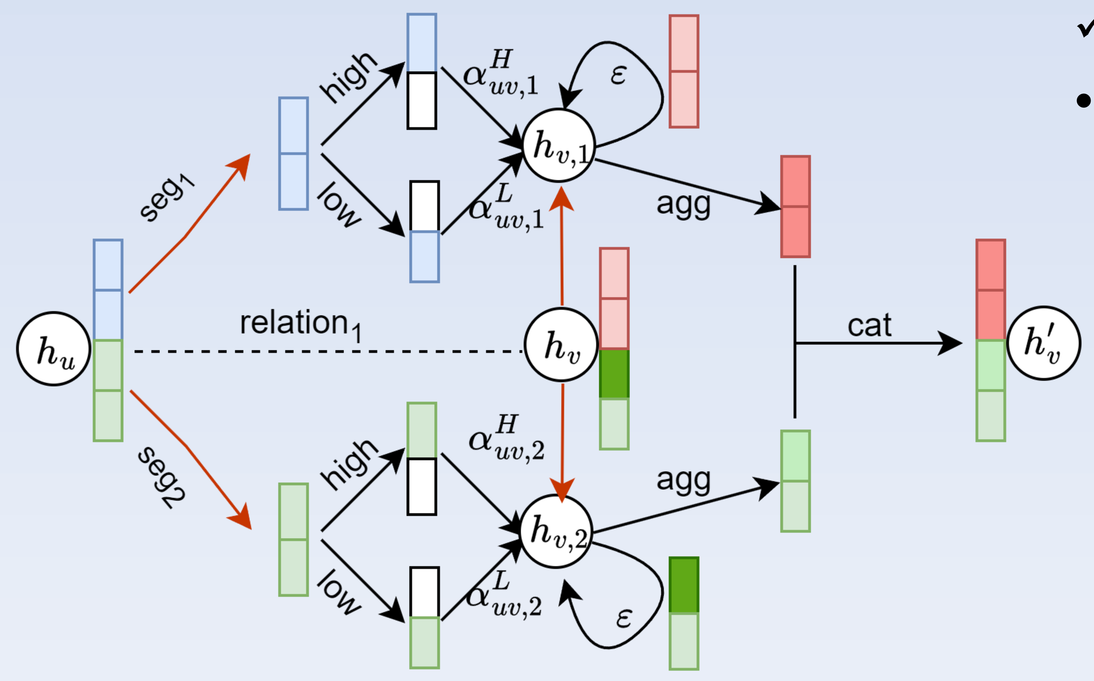
Main Framework -- F²GNN

✓ Feature Segmentation

- We transform original node features into higher-dimensional embeddings, then divide these into S_n segments, each of dimension S_d.

$$h_i^{(0)} = \sigma(W_s x_i)$$

$$h_i^{(0)} = \Big\Vert_{k=1}^{S_n} h_{i,k}^{(0)}$$



✓ Adaptive Filter Analysis

- We simultaneously use high-frequency and low-frequency filters and linearly combine them for adaptive filtering.

$$\mathcal{F}_L = \varepsilon I + D^{-1/2} A D^{-1/2}$$

$$\mathcal{F}_H = \varepsilon I - D^{-1/2} A D^{-1/2}$$

$$h_i' = \alpha_{ij}^L (\mathcal{F}_L \cdot \mathbf{H})_i + \alpha_{ij}^H (\mathcal{F}_H \cdot \mathbf{H})_i = \varepsilon h_i + \sum_{j \in N(i)} \frac{\alpha_{ij}^L - \alpha_{ij}^H}{\sqrt{d_i d_j}} h_j$$

Experimental Comparison

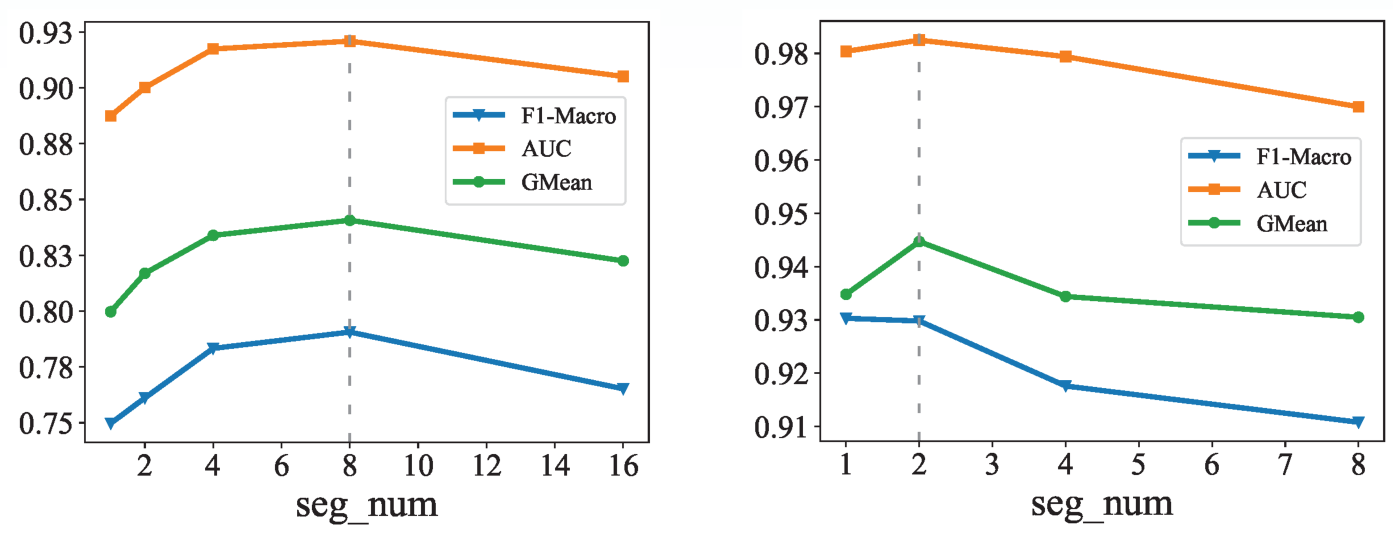
Method	Dataset	YelpChi			Amazon		
	Metric	F1-macro	AUC	GMean	F1-macro	AUC	GMean
Baselines	GCN	0.4979±0.0008	0.5611±0.0005	0.5141±0.0017	0.6625±0.0011	0.8173±0.0009	0.6801±0.0022
	GAT	0.5204±0.0059	0.5703±0.0023	0.5121±0.0184	0.6790±0.0025	0.8308±0.0035	0.6812±0.0039
	GraphSAGE	0.5781±0.0239	0.7409±0.0034	0.6815±0.0049	0.8383±0.0109	0.9149±0.0077	0.8518±0.0077
	CARE-GNN	0.6281±0.0137	0.7918±0.0002	0.7279±0.0035	0.8765±0.0011	0.9425±0.0156	0.8982±0.0015
	PC-GNN	0.6240±0.0665	0.8500±0.0147	0.7543±0.0322	0.8820±0.0053	0.9664±0.0054	0.9095±0.0056
	AO-GNN	0.7042±0.0051	0.8805±0.0008	0.8134±0.0232	0.8921±0.0045	0.9640±0.0020	0.9096±0.0105
	H^2-FDetector	0.7301±0.0014	0.8999±0.0041	0.8232±0.0047	0.8480±0.0433	0.9391±0.0235	0.9108±0.0179
	NGS	0.7754±0.0048	0.9134±0.0021	0.8244±0.0069	0.9202±0.0020	0.9720±0.0043	0.9213±0.0065
	BWGNN(Homo)	0.7311±0.0032	0.8513±0.0069	0.7677±0.0075	0.9181±0.0057	0.9745±0.0035	0.9266±0.0038
	BWGNN(Hetero)	0.7895±0.0044	0.9130±0.0048	0.8299±0.0079	0.9159±0.0061	0.9764±0.0024	0.9213±0.0072
	BHomo-GHRN	0.7312±0.0062	0.8599±0.0063	0.7747±0.0071	0.9199±0.0058	0.9643±0.0090	0.9112±0.0120
	BHetero-GHRN	0.7751±0.0092	0.9077±0.0053	0.8282±0.0069	0.9151±0.0105	0.9706±0.0040	0.9188±0.0079
Ablation	$F^2GNN_{\backslash A}$	0.7248±0.0029	0.8634±0.0037	0.7687±0.0190	0.8843±0.0072	0.9709±0.0007	0.9105±0.0045
	$F^2GNN_{\backslash S}$	0.7497±0.0251	0.8874±0.0134	0.7997±0.0230	**0.9303±0.0051**	0.9814±0.0016	0.9359±0.0059
Ours	F^2GNN	**0.7907±0.0051**	**0.9206±0.0037**	**0.8317±0.0086**	0.9278±0.0042	**0.9825±0.0011**	**0.9447±0.0029**

- The feature segmentation of F²GNN gains more improvement on YelpChi than Amazon. Because YelpChi has a larger number of fraudsters, but the similarity of node features is high, indicating that fraudulent features are more concealed. In contrast, Amazon exhibits more obvious fraud features.

Sensitivity

- The optimal performance is achieved when the features are divided into 8 segments on YelpChi, while on Amazon, the best results are obtained with 2 segments.
- YelpChi has more concealed fraud features, so it requires more segments to increase the granularity.
- Excessive segmentation may introduce unnecessary complexity and decrease model performance.

(a) YelpChi (b) Amazon



Conclusion

- This paper studies the adversarial camouflage of fraudulent features and class-imbalance issues in graph-based fraud detection.
- We introduce a novel approach of feature segmentation modeling and apply frequency-adaptive filters on the corresponding segments. They are used respectively to address fraud camouflage and class imbalance issues between fraudulent and benign users.
- The experiments on two public real-world datasets validate the effectiveness of our proposed method.

Contact

Scan the QR code to get the paper, code, and author contact. Any questions and suggestions are welcomed.

Email: liuyang2023@ict.ac.cn, liuyang520ict@gmail.com

➢ Paper: ➢ Code: ➢ Homepage: ➢ Wechat: