NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

IARIA

Trigger Injection via Clustering for Backdoor Attacks on Heterogeneous Graphs

Honglin Gao, Lan Zhao, Gaoxi Xiao

School of Electrical and Electronic Engineering, Nanyang Technological University(Singapore)
Contact Email: **HONGLIN001@e.ntu.edu.sg**

Honglin Gao

Honglin Gao is currently a Ph.D. candidate at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. He received his B.Eng. in Internet of Things Engineering from Beijing University of Posts and Telecommunications, and his M.Sc. in Artificial Intelligence from NTU's College of Computing and Data Science in 2022.

His research interests include graph neural networks, heterogeneous graph modelling, and adversarial attacks.

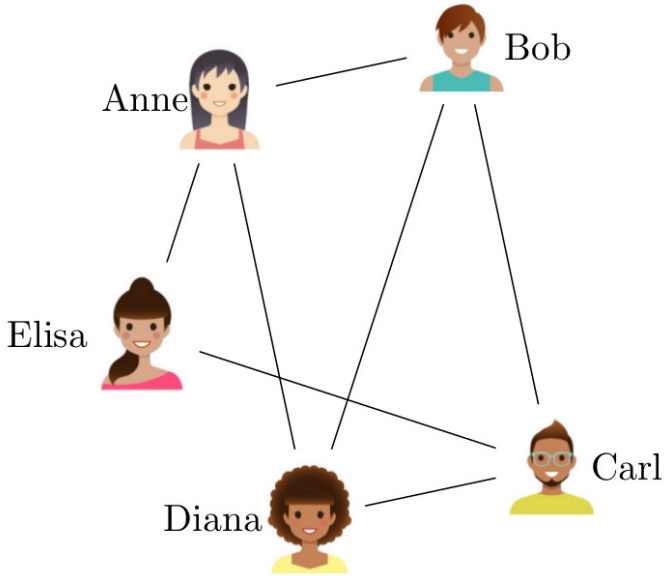
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Research Interests

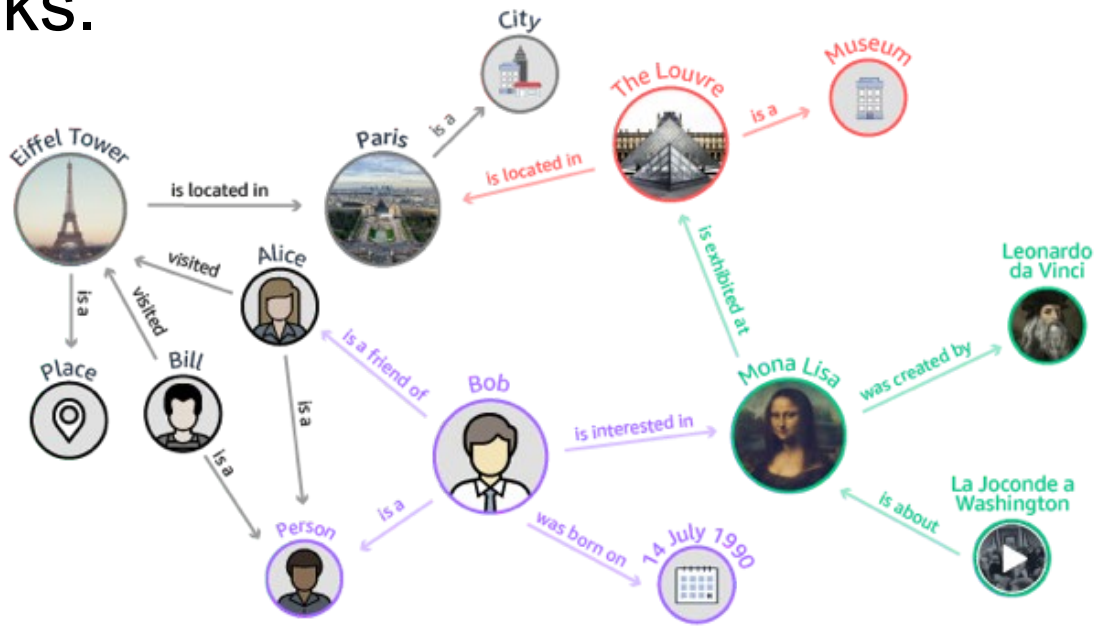
- Complex systems and complex networks
- Cyber-physical systems
- Cyber-physical security
- Resilience engineering
- Optimization algorithms and applications
- Internet and communication networks

Research Background

- Graph-structured data have widespread applications in social networks, finance, and biology.
- HGNNs leverage multi-typed nodes and edges to capture richer semantics compared to traditional GNN, leading to better performance on heterogeneous tasks.



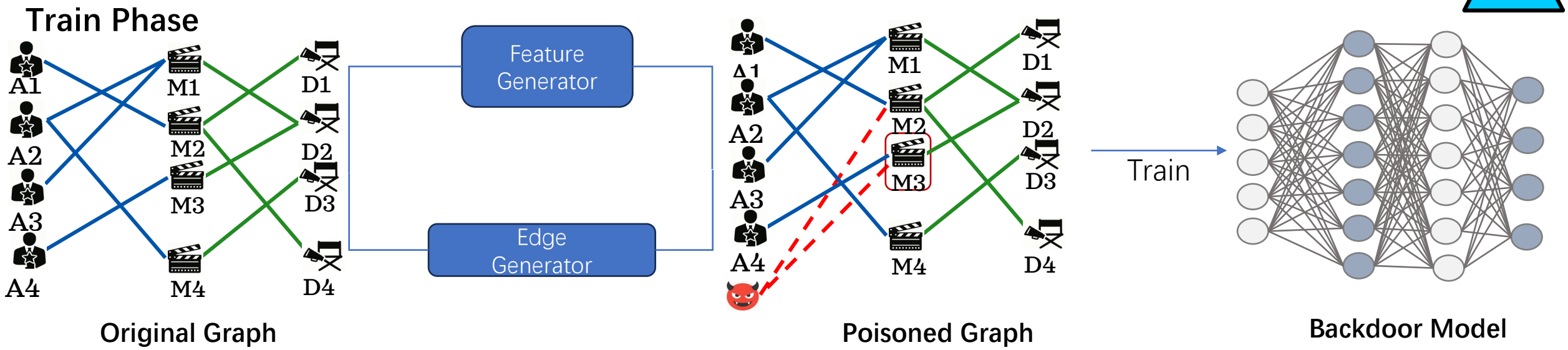
Traditional Graph

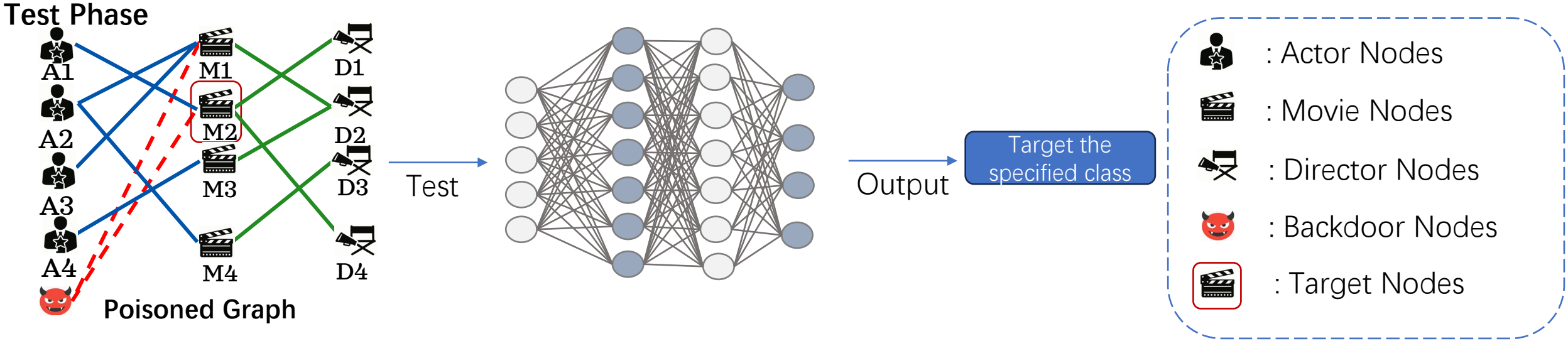


Heterogeneous Graph

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

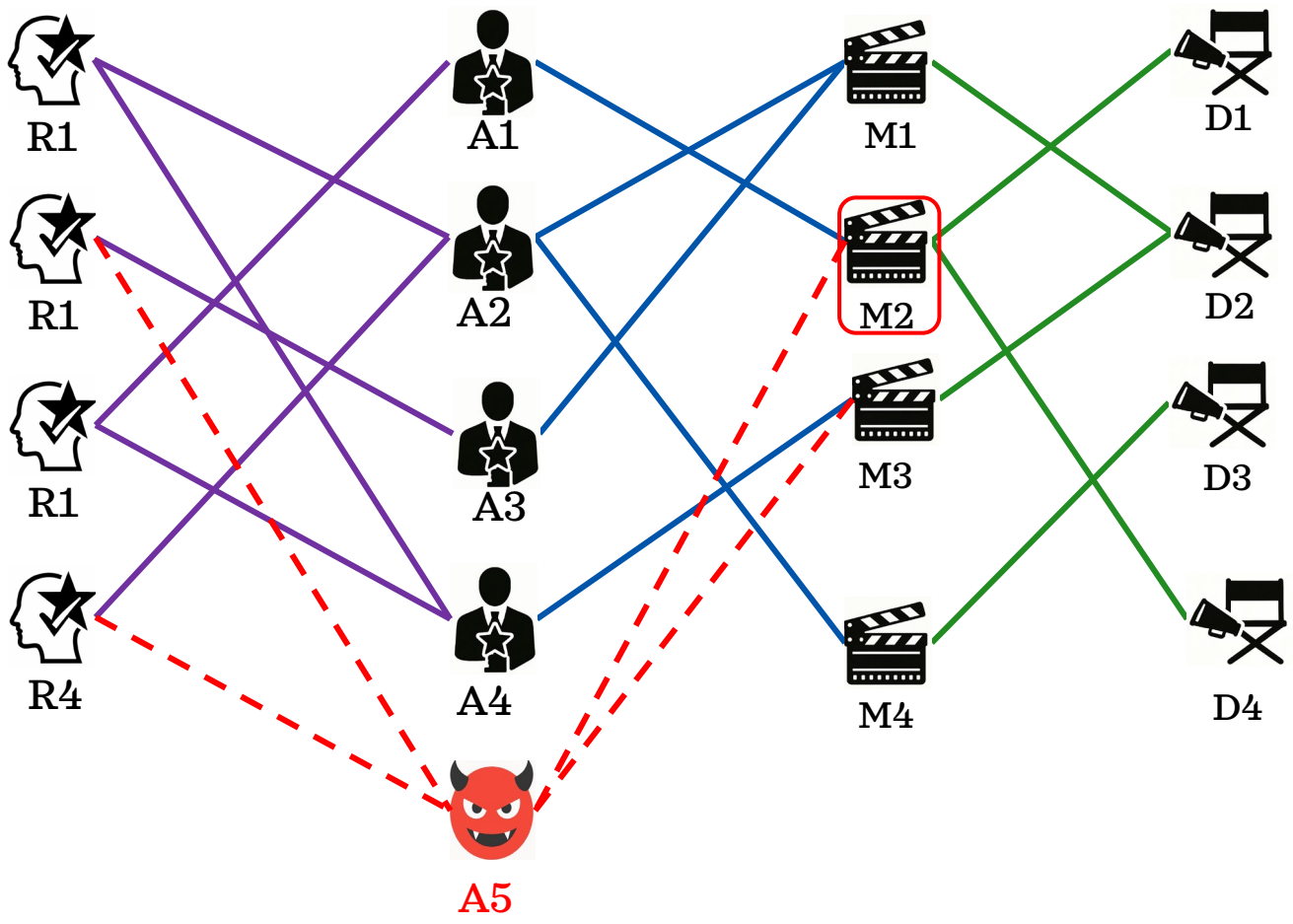
Research Background





NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Preliminaries



: Primary Type

M2

: Primary Nodes

: Trigger Type

A5

: Trigger Nodes

: Auxiliary Type

R1, R4 M3

: Auxiliary Nodes

Problem Formulation

Problem Setting

Input: A heterogeneous graph $G = (\mathcal{V}, \mathcal{E}, X)$ with node types T and edge types R

Task: Classify nodes of the primary type t_p

Attacker's Goal:

- Inject trigger nodes $\mathcal{V}_{t_{tr}}^{new}$ along with their features X^{new} and edges $\mathcal{E}^{new}$ into the graph.
- Maximize attack success while minimally affecting clean nodes.

Formal Objective

Attack Objective:

$$\widetilde{G^*} = \arg\max_{\tilde{G} \in \mathcal{F}(G)} \left[\sum_{v \in \mathcal{V}^{(p)}} 1\left(f_\theta(\tilde{G}, v) = y_t\right) + \sum_{v \in \mathcal{V}_{t_p} \setminus \mathcal{V}^{(p)}} 1\left(f_\theta(\tilde{G}, v) = y_v\right) \right]$$

where:

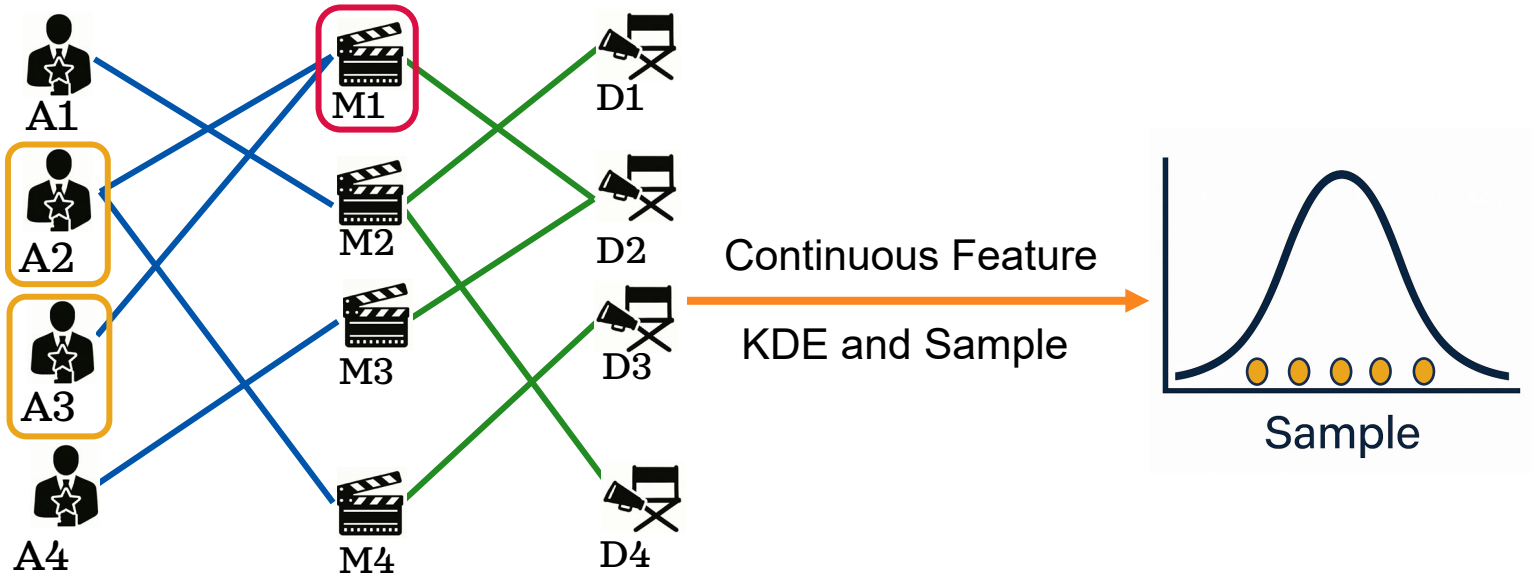
$$\tilde{G} = \left(\tilde{V}, \tilde{E}, \tilde{X}\right), \tilde{V} = V \cup V_{t_{tr}}^{new}, \tilde{\mathcal{E}} = \mathcal{E} \cup \mathcal{E}^{new}, \tilde{X} = \begin{bmatrix} X \\ X^{new} \end{bmatrix}$$

$\mathcal{V}^p$: Set of poisoned nodes
$1(\cdot)$: Indicator function
X^{new}: Feature matrix of injected nodes
$\mathcal{F}(G)$: Space of allowed graph modifications
$\mathcal{E}^{new}$: Edges from injected nodes

Feature Generator

Data Extraction
Collect 1-hop trigger-type neighbours from target-class nodes

Continuous Feature Generation
Fit a KDE and sample from the distribution



Target Class Node

1-hop trigger-type neighbours

Edge Generator

Core Idea: Select auxiliary nodes with the highest average embedding similarity to ensure that trigger connections align with key regions of the graph.

Identify Target Nodes

Select all primary-type nodes labelled as the target class y_t, denoted as V_{y_t}

Locate 1-hop Trigger-type Neighbors

For each target node $v_{yt} \in V_{yt}$ retrieve its 1-hop neighbors of the trigger type through edge relation $r_{t_p,t_{tr}}$ forming the neighbour set $V_{t_{tr}}^{(1)}(v_{y_t})$

Locate 2-hop Auxiliary-type Candidates

Identify 2-hop auxiliary nodes $V_{t_{aux}}^{(2)}$ through $V_{t_{tr}}^{(1)}$ and edges $r_{t_{tr},t_b}, t_b \in T_{aux}$

Extract Node Embeddings

Extract the node embeddings $\boldsymbol{z_v}$ for all $v \in V_{aux}^{(2)}$ using a surrogate model

Calculate Clustering Scores

Compute the influence score base on embeddings $\mathrm{I}\left(v_{\mathrm{aux}}^{(2)}\right)$ with cosine

Select Top Auxiliary Nodes

Rank auxiliary nodes by $\mathrm{I}\left(v_{\mathrm{aux}}^{(2)}\right)$ select the top-d_{t_b} nodes for each auxiliary

Connect to Trigger Nodes

Connect each newly inserted trigger node to the selected top auxiliary nodes

Influence Score Calculation: $\mathrm{I}\left(\mathrm{v}_{\mathrm{aux}}^{(2)}\right) = \frac{1}{\left|\mathrm{v}_{\mathrm{aux}}^{(2)}\right|-1} \sum_{v' \in V_{\mathrm{aux}}^{(2)}, v' \neq v} \left(\frac{z_v}{|z_v|_2} \cdot \frac{z_{v'}}{|z_{v'}|_2}\right)$

Experiments

Table I
BACKDOOR ATTACK EFFECTIVENESS ON IMDB DATASET

Dataset	Victim Model	Class	Trigger	ASR				CAD			
				HeteroBA-C	HeteroBA-R	CGBA	UGBA	HeteroBA-C	HeteroBA-R	CGBA	UGBA
IMDB	HAN	0	director	**0.9953**	0.6791	0.5618	0.2087	0.0307	0.0265	**0.0037**	0.0364
		1		**0.9984**	0.8458	0.4523	0.2991	-0.0031	-0.0094	**-0.0119**	0.0037
		2		**1.0000**	0.9003	0.4992	0.3582	0.0068	**-0.0068**	0.0010	0.0067
	HGT	0	director	**0.8473**	0.7975	0.4851	0.5109	0.0036	0.0021	**-0.0104**	0.0291
		1		**0.9299**	0.8878	0.4147	0.7757	0.0182	−0.0146	**0.0130**	0.0026
		2		**0.8894**	0.8193	0.4523	0.6807	0.0026	**-0.0099**	-0.0015	0.0182
	SimpleHGN	0	director	**0.9533**	0.7679	0.3881	0.8443	-0.0047	0.0015	**-0.0244**	0.0005
		1		0.9502	0.9486	0.3850	**0.9595**	0.0047	0.0052	**-0.0130**	0.0291
		2		**0.9720**	0.8255	0.3474	0.9330	**-0.0052**	−0.0166	0.0156	0.0078

$$ASR = \frac{\sum_{i=1}^{n} \mathbf{1}(f_b(v_i) = y_t)}{n}$$

n: Number of target nodes under attack
f_b:Backdoor Model
y_t:Attacker specified target label

$$CAD = Acc_{f_c}(\text{Clean}) - Acc_{f_b}(\text{Clean})$$

$Acc_{f_c}(clean)$: Accuracy of the clean model on clean data
$Acc_{f_b}(clean)$: Accuracy of the backdoor model on clean data

Conclusion and Future Work

Conclusion

- Proposed HeteroBA, a structure-based backdoor attack for heterogeneous graphs.
- Uses feature and edge generators to insert semantically valid trigger nodes.
- Achieves high attack success rate (ASR) with minimal clean accuracy drop (CAD).
- Demonstrates strong stealthiness and reveals security risks of HGNNs.

Future Work

- Extend to more datasets (e.g., DBLP, Amazon, OAG).
- Improve scalability with sampling-based clustering and mini-batch KDE.
- Test on diverse victim models (e.g., GAT, Transformer-based HGNNs).
- Develop adaptive defense mechanisms against structure-aware backdoors.

Reference List

[1] A. Salamat, X. Luo, and A. Jafari, “Heterographrec: A heterogeneous graph-based neural networks for social recom mendations”, Knowledge-Based Systems, vol. 217, p. 106817, 2021.

[2] D. Singh and A. Verma, “An overview of heterogeneous social network analysis”, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 15, no. 2, e70028, 2025.

[3] S. Xiang, D. Cheng, C. Shang, Y. Zhang, and Y. Liang, “Tempo ral and heterogeneous graph neural network for financial time series prediction”, in Proceedings of the 31st ACM international conference on information & knowledge management, 2022, pp. 3584–3593.

[4] E. Dai, M. Lin, X. Zhang, and S. Wang, “Unnoticeable backdoor attacks on graph neural networks”, in Proceedings of the ACM Web Conference 2023, 2023, pp. 2263–2273.

[5] X. Xing, M. Xu, Y. Bai, and D. Yang, “A clean-label graph backdoor attack method in node classification task”, Knowledge Based Systems, vol. 304, p. 112433, 2024.

[6] Y.-C. Chen, “A tutorial on kernel density estimation and recent advances”, Biostatistics & Epidemiology, vol. 1, no. 1, pp. 161 187, 2017.

[7] X. Wang et al., “Heterogeneous graph attention network”, in The world wide web conference, 2019, pp. 2022–2032.

[8] X. Fu, J. Zhang, Z. Meng, and I. King, “Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding”, in Proceedings of the web conference 2020, 2020, pp. 2331–2341.

[9] X. He et al., “Lightgcn: Simplifying and powering graph convolution network for recommendation”, in Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.

[10] W. Hu et al., “Open graph benchmark: Datasets for machine learning on graphs”, Advances in neural information processing systems, vol. 33, pp. 22118–22133, 2020.

[11] P. Veli˘ckovi´c et al., “Graph attention networks”, arXiv preprint arXiv:1710.10903, 2017.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model”, IEEE transactions on neural networks, vol. 20, no. 1, pp. 61–80, 2008.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Thanks for your attention!

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE