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
e-mail: HONGLIN001@e.ntu.edu.sg, zhao0468@e.ntu.edu.sg, egxxiao@ntu.edu.sg

***Abstract*—Heterogeneous graph neural networks have achieved remarkable success in modeling multi-relational data. However, the risks associated with backdoor attack have largely gone unexplored. In this paper, we present a new structure-based backdoor attack method for heterogeneous graph neural networks. Our method uses a set of designed trigger nodes in the graph connected to semantically related parts of the graph using clustering-based trigger node selection. Triggering nodes cause the model to misclassify certain target nodes as an attacker-specified class while still keeping a high accuracy on the clean data. Preliminary experiments on publicly available benchmark datasets show that our proposed backdoor attack is effective and stealthy. This shows that there is a clear need for security awareness in heterogeneous graph learning.**

Keywords-heterogeneous graph; backdoor attack.

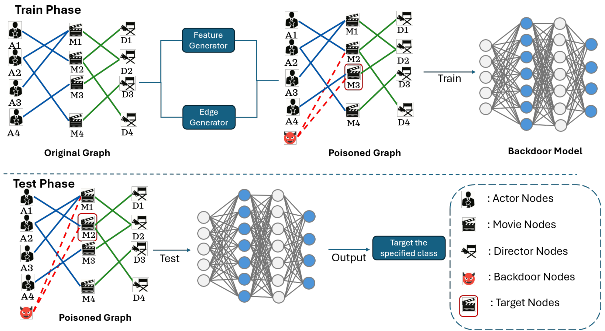

Figure 1. Backdoor process.

I. Introduction

Heterogeneous Graph Neural Networks (HGNNs) have quickly become prominent for leveraging multi-typed relational data, such as through recommendation [1], social analysis [2] and financial intelligence applications [3]. While HGNNs have gained much success in applications, there has been a lack of investigation into their security. Just like their homogeneous counterparts, HGNNs are susceptible to backdoor attacks; intentional corruption of a model such that the model will perform incorrectly when using a certain trigger. Backdoor attacks are serious attacks that have been overlooked by many researchers.

Unlike traditional adversarial attacks, backdoor attacks implant a hidden pattern during training that causes abnormal responses to specific triggers. While such attacks can target various graph-based tasks, this work focuses on the classification setting, where at inference time the model behaves normally on clean data but misclassifies target nodes into attacker-specified classes when the input contains the trigger.

Numerous notable approaches have been proposed for homogeneous graph backdoor attack, such as Unnoticeable Graph Backdoor Attack (UGBA) [4] and Clean-label Graph Backdoor Attack (CGBA)[5]. Specifically, UGBA employs structure-level triggers by optimizing the triggering structure's topological similarity with benign substructures, with the goal of minimizing the visibility of perturbation, and avoiding structural detection. Alternatively, CGBA utilizes a clean-label approach by injecting feature-based triggers into nodes belonging to the target class, without any modifications to the labels or graph structure. In general, both methods assume same node types with homogeneous edge semantics, and lack modeling mechanisms to adequately represent semantic constraints in heterogeneous graphs, such that their overall attack effectiveness cannot be fully realized.

Our method achieves target-specific misclassification without compromising the overall performance of the model. This is accomplished by adding carefully selected trigger nodes and forming edges that are consistent with the types of the key regions. The preliminary results suggest that these assaults are highly effective and difficult to detect, raising concerns over the security of HGNN-based systems. This method has several problems because it needs the capacity to change the network topology and a complete understanding of graph schemas, which makes it less useful in black-box situations.

The remainder of this extended abstract is organized as follows. Section II introduces the proposed backdoor attack framework designed for heterogeneous graphs. Section III presents experimental results on the IMDB dataset(a comprehensive online databases of movies, TV shows, actors, and production crew information), which shows the relationships between movies, actors and directors, to validate the effectiveness and stealthiness of the attack. Section IV provides a comparative discussion with existing methods, and Section V concludes the paper with future directions.

II. Methods

We propose Heterogeneous Backdoor Attack (HeteroBA), a structure-manipulating backdoor attack framework tailored for heterogeneous graphs. The core idea is to insert a node of a type that can legally connect to the target node type, and generate semantically coherent features for it using a Feature Generator. To provide coherence with the relational constraints of heterogeneous graphs, an Edge Generator links this trigger node to other present nodes in a type-consistent manner. This

Copyright (c) IARIA, 2025.   ISBN: 978-1-68558-293-7

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

enables the injected node to propagate misleading information to the target node while maintaining high stealthiness.

The overall attack pipeline is illustrated in Figure 1. The Feature Generator and Edge Generator work together to insert a crafted trigger node (e.g., the red node) into the graph, connecting it to semantically relevant regions. During training, the trigger is embedded into the model without degrading clean performance. At test time, when the same structural pattern reappears and connects to a target node, it activates the backdoor behavior, causing the target to be misclassified into the attacker-specified class.

In order to provide stealth for the implanted trigger nodes, the Feature Generator gives them feature vectors with properties close to those of benign nodes of the same type. This is done by modeling the feature distribution in relation to the trigger node type using Kernel Density Estimation (KDE) [6]. KDE is a non-parametric method of estimating the probability density function of a random variable. In our case, it captures the empirical feature distribution of clean nodes from the target class.

For a given set of clean nodes of a certain type (e.g., "author" nodes in an academic network), we apply Kernel Density Estimation (KDE) to get a smoothed estimate of their feature space. We then sample new feature vectors for trigger nodes from this estimated distribution. This method guarantees that the generated features are statistically indistinguishable from those of valid nodes, thereby rendering it hard to identify trigger nodes via feature-based anomaly detection methods.

To enhance the influence of the trigger and improve its stealthiness within the graph structure, the Edge Generator in HeteroBA adopts a clustering-based strategy to determine how the trigger node is connected. Specifically, we first identify a subset of nodes that are legally allowed to connect to the target node type according to the schema of the heterogeneous graph. We then perform clustering within this subset based on node feature information, dividing the candidates into several semantically coherent and structurally compact regions.

After clustering, we select some influential nodes and connect the trigger node to them. Not only does this approach ensure legitimate edge types, but it also inserts the trigger into a meaningful local context. Compared to random, clustering-driven edge selection improves the attack effectiveness while preserving the graph's overall structure, making the backdoor harder to detect.

III. Results

We evaluate our method on the widely used IMDB dataset, which is a heterogeneous graph composed of three node types: movies, directors, and actors, with edges representing semantic relations such as directed-by and acted-in. In our attack setting, director nodes are injected as trigger nodes to manipulate the classification results of movie nodes. A visual comparison of the graph before and after trigger injection is presented in Figure 1.

To measure attack effectiveness, we employ two conventional evaluation metrics. The Attack Success Rate (ASR) is the ratio of poisoned target nodes that are misclassified into an attacker-chosen label at inference. The Clean Accuracy Drop (CAD) shows how much test accuracy goes down on clean data, which shows how stealthy the attack is [4].

As shown in Table I, under the HAN model [7], our proposed method HeteroBA-C (which uses clustering-based edge injection) achieves over 99% ASR across all target classes, significantly outperforming baselines such as CGBA and UGBA. The CAD, on the other hand, stays within ±0.01, which means that clean data is not affected much.

To further validate the effectiveness of the clustering-based edge injection strategy, we compare HeteroBA-C with a variant called HeteroBA-R, in which the injected trigger node connects to randomly selected legal-type nodes instead of semantically coherent clusters. Table I shows that HeteroBA-R has a much lower ASR, while CAD is similar to HeteroBA-C. This contrast shows that clustering-based structural placement greatly improves the effectiveness of attacks without sacrificing stealthiness.

IV. Discussion

The comparison of the IMDB dataset indicates that HeteroBA works far more effectively when dealing with graphs that have multiple types of nodes and connections. By incorporating type-compatible trigger nodes and semantically consistent edges, our approach attains better attack efficacy with minimal disturbance to accurate predictions.

In terms of computational cost, the dominant overhead of HeteroBA lies in the clustering-based auxiliary node selection. Let p denote the number of target nodes and n_{aux} the number of auxiliary nodes. For each target node, HeteroBA performs a clustering operation with complexity $O(n_{\mathrm{aux}} \log n_{\mathrm{aux}})$, resulting in an overall time complexity of $O(p \cdot n_{\mathrm{aux}} \log n_{\mathrm{aux}})$.

Other steps such as feature sampling and edge insertion incur negligible cost. This demonstrates that HeteroBA balances both attack performance and computational scalability, making it feasible for practical use in real-world heterogeneous graphs.

CGBA, on the other hand, only foucuses on feature-level modification by finding the most discriminative feature dimension and using it as a trigger in the poisoned nodes. This method is simple, it lacks structural adaptability. More importantly, in real-world applications such as social networks or recommendation systems, directly altering node features (e.g., modifying user profiles or item attributes) is often impractical or easily detectable. HeteroBA's method of adding additional nodes or edges, on the other hand, is both achievable and undetected, making it easier to fit into existing graph structures.

UGBA, on the other hand, employs a bi-level optimization strategy: The inner loop increases the classification confidence of the poisoned nodes, while the outer loop uses cosine distance to make sure that the features are similar at the feature level. This design works well in homogeneous environments, but it does not quite capture the complex semantics of different node types and relationships found in heterogeneous graphs. As a result, its triggers lack contextual compatibility, reducing both effectiveness and stealth.

The consistently low CAD values across multiple classes and models confirm that HeteroBA is not too noticeable. These findings clearly demonstrate that HGNNs are particularly susceptible to backdoor attacks that exploit their structural awareness. They stress the urgent need to create targeted protection mechanisms that are tailored to the specific semantics and realistic structures present in these models.

V. Conclusion and Future Work

In this work, we propose HeteroBA, a heterogeneous graph neural network backdoor attack framework that perturbs structure specifically crafted for the task. By co-designing feature and edge generators according to the graph schema, HeteroBA is able to inject semantically plausible triggers that cause targeted misclassification with minimal negative impact on clean data. Experiments on the IMDB dataset validate its high attack success rate and remarkable stealth capabilities over baselines with demonstrated performance.

For future work, We intend to expand HeteroBA toa broader range of heterogeneous graph datasets in different domain, including academic networks (e.g., DBLP-a computer science bibliography website) [8], e-commerce networks (e.g., Amazon) [9], and extensive bibliographic graphs (e.g., OAG-a large knowledge graph unifying two billion-scale academic graphs) [10]. To solve scalability problems in such large graphs, we will look into more efficient versions of our approach, including clustering with sampling or mini-batch KDE-based feature generation, to reduce the amount of computing resources needed without lowering performance. We also intend to evaluate our method under more diverse victim models, including Graph Attention Networks (GAT) based [11] and transformer-based heterogeneous Graph Neural Networks (GNNs) [12]. In addition, we aim to explore adaptive defense mechanisms capable of detecting or neutralizing structure-aware backdoors in heterogeneous settings.

Acknowledgement

The authors gratefully acknowledge the partial support of the Ministry of Education, Singapore, under the research grant RG10/23.

References

[1] A. Salamat, X. Luo, and A. Jafari, "Heterographrec: A heterogeneous graph-based neural networks for social recommendations", *Knowledge-Based Systems*, vol. 217, p. 106 817, 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, "Temporal 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. 112 433, 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. 22 118–22 133, 2020.

[11] P. Veličković *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.