Understanding the Influence of Extreme High-degree Nodes in Graph Anomaly Detection

Authors

Institutes

Abstract. Graph Anomaly Detection (GAD) has attracted considerable attention for its potential in detecting anomalies. However, an overlooked issue in prior research is the presence of extreme high-degree node. Extreme high-degree node plays a pivotal role in GAD, yet its influence remains uncertain due to the absence of suitable GAD datasets. To address this gap, this paper first introduces a novel graph anomaly dataset, NFTGraph, which incorporates extreme high-degree nodes. Subsequently, various experiments are conducted on this dataset to elucidate the influence of extreme high-degree nodes on GAD, encompassing aspects such as model backbone, computational costs, and the phenomenon of over-smoothing, among others. Finally, to mitigate the influence, we propose a new algorithm, SNGNN. Experimental results illustrate that SNGNN significantly surpasses existing models, yielding an average detection AUROC improvement of over 2%, thereby notably enhancing anomaly detection performance.

Keywords: Graph Anomaly Detection · Extreme High-degree Nodes · Graph Construction.

1 Introduction

Graph, a data structure with nodes and edges, has been widely used to model real-world scenarios, such as social networks [6], financial trading networks [16] and paper citing networks [5]. Since graphs can capture relationships between entities, many anomaly detection methods are also based on graphs [1,4], aiming to identify anomalies that are distinct from the majority in the graph. Historically, numerous models for graph anomaly detection (GAD) have been put forth, such as DOMINANT [4], CONAD [19], and PCGNN [10]. These models have significantly contributed to the advancement of GAD.

However, a critical aspect overlooked by prior GAD studies pertains to the existence of nodes with exceptionally high degrees in the graph, such as influential nodes in social networks and financial institutions in trading networks. As demonstrated in Sec.4, extreme high-degree nodes may introduce noise through neighbor aggregation and result in inadequate modeling of anomalous nodes. Regrettably, previous graph anomaly datasets did not encompass extreme high-degree nodes. As illustrated in Figure1, several widely-utilized GAD datasets, including Amazon [12] and Yelpchi [14], exhibit subtly distributed node degrees.

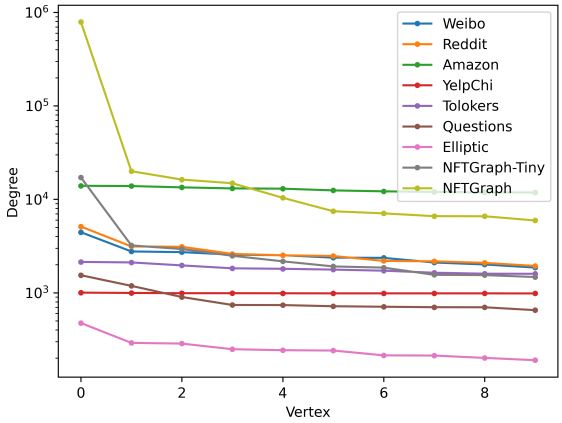


Fig. 1: The distribution of node degrees among the foremost ten nodes.

The use of injected or synthetic graph datasets is unfeasible due to the potential risk of significant data leakage [7]. These challenges hinder the advancement of GAD models in addressing this pivotal concern.

To address the aforementioned limitation, we introduce a novel graph anomaly dataset named NFTGraph, which includes extreme high-degree nodes. Illustrated by the yellow and gray lines in Figure 1, NFTGraph and its derived variant, NFTGraph-Tiny, exhibit the most pronounced slope between the first and second nodes. Leveraging this provided graph anomaly dataset, we conduct a series of experiments to examine the influence of extreme high-degree nodes and derive insightful conclusions. To mitigate the influence of these nodes, we propose SNGNN. Experimental results demonstrate that the algorithm significantly outperforms existing models, yielding an average detection AUROC improvement of over 2%, thereby greatly enhancing the performance of anomaly detection.

In summary, our contributions are as follows:

- We propose NFTGraph, a graph anomaly dataset derived from blockchain data, characterized by extreme high-degree nodes.
- A series of experiments are conducted on the proposed NFTGraph to comprehend the influence of extreme high-degree nodes. Meaningful insights are drawn, underscoring the potential for advancing graph anomaly detection.
- SNGNN is introduced to alleviate the influence of extreme high-degree nodes. Experimental results demonstrate that SNGNN surpasses existing methods across four datasets, yielding an average improvement of over 2% in detection AUROC.

Moreover, both the dataset and codes have been made publicly accessible on Github[1].

[1] https://github.com/AnonymousDataCodeHub/.

2 Related Works

Graph Anomaly Datasets: Numerous graph anomaly datasets are widely employed in research endeavors. For example, Weibo [8] and Reddit [8], renowned datasets derived from social networks designed for the detection of suspicious users. Questions [13] is a question-answering dataset employed for detecting active users. While these classical datasets have significantly contributed to the development of graph anomaly detection, they lack extreme high-degree nodes showcased in Figure 1. Thus, they are not adequately qualified for exploration.

Degree-related GNNs: Historically, several GNNs with a focus on degree-related considerations have been introduced to rectify node degree distribution biases. Notable examples include DEMO-Net [18] and SL-DSGCN [17], which implement degree-specific node transformations, and DegFairGNN [11], which employs a learnable function for generating debiasing contexts. However, these models have primarily been explored within the context of node or graph classification tasks. Thus far, a scarcity of research has addressed the ramifications of extreme high-degree nodes in anomaly detection tasks.

3 NFTGraph Construction and Properties

The construction of NFTGraph involves the following steps: (1) **Raw data:** The blockchain is essentially a ledger, where each transaction is documented. We extract certain fields of ERC-1155 NFT transaction on the Ethereum blockchain to compose the format of raw data (Table 1). (2) **Graph Structure:** The *From* and *To* addresses, acting as the sending and receiving parties of a transaction, serve as the source and target nodes in the graph. An edge is established between the source and target nodes if tokens are transferred between them. (3) **Node Feature:** Utilizing the transaction fields, node features are constructed. Each node possesses 50-dimensional attributes, encompassing the factors of Degree, Amount, Value, and TxFee in both outbound and inbound directions. These features are derived through various operators, including `sum`, `mean`, `max`, `min`, `first` (earliest transaction), `last` (latest transaction), `median`, and `nunique` (count of distinct ones). (4) **Labeling Suspicious Node:** First, we define *ground-truth fraudulent nodes* that are the account addresses involved in a variety of fraudulent schemes published by previous researchers, encompassing Ponzi schemes [3] and phishing scams [2]. Then, we label nodes that exhibit interactions with the fraudulent nodes exceeding a count of three instances as *suspicious nodes*. Suspicious nodes aim to alleviate the notable imbalance, arising from the limitation of ground-truth fraudulent nodes aligning with the NFTGraph's node set. (5) **Variant Dataset:** By extracting 20,000 of the most active nodes while excluding isolated nodes, we form NFTGraph-Tiny, leading to a substantial size reduction. This is executed with the recognition that certain GNNs may encounter challenges in handling extensive graphs within resource-constrained environments, a scenario often encountered in GAD.More details of NFTGraph construction are described in the supplemental materials.

Table 1: Format of raw NFT transaction data.

TxHash	*From*	*To*	*Token*	*Timestamp*	*Amount*	*Value*($)	*TxFee*($)
0xb5...b420	0x94...7293	0x6e...b7d3	0xd0...2430	20220730055230	1	78.52	2.23
0xa5...aeea	0x00...0000	0xd8...ac95	0xd0...2430	20220730055230	14	0.0	0.98
0xa2...bdf1	0x5b...1abb	0x4f...6580	0xd0...2430	20220730055138	1	0.0	0.33
...	...	...	...	...	...	...	...

Properties: Table 2 illustrates a comparison of statistical properties between NFTGraph and several other graph anomaly datasets [15]. The original NFTGraph comprises 1,161,847 nodes and 2,851,407 edges, constituting a large-scale dataset. NFTGraph-Tiny represents the core of the NFT transaction network, containing only 20,000 nodes and 245,221 edges. The anomaly ratio of NFTGraph is only 0.39%, making it the lowest in the dataset, presenting a significant challenge for detection models due to this extreme imbalance. Moreover, in NFTGraph, the highest degree (No.1 deg) is 789,782, significantly surpassing No.2 deg. This pattern is consistent in NFTGraph-Tiny, but in other graph datasets, the discrepancy between No.1 deg and No.2 deg is less pronounced.

Table 2: Statistics of NFTGraph and some common graph anomaly datasets

Dataset	#Nodes	#Edges	Anomaly	Avg deg	No.1 deg	No.2 deg	q_1	q_2
Weibo	8,405	416,368	10.3%	99.08	4,447	2,769	44.88	27.95
Reddit	10,984	168,016	3.3%	30.59	5,112	3,134	167.10	102.44
Amazon	11,944	8,847,096	9.5%	1481.43	13,964	13,874	9.43	9.37
YelpChi	45,954	7,739,912	14.5%	336.85	1,004	996	2.98	2.96
Tolokers	11,758	530,758	21.8%	90.28	2,140	2,113	23.70	23.40
Questions	48,921	202,461	3.0%	8.28	1,541	1,186	186.18	143.29
Elliptic	203,769	438,124	9.8%	4.30	475	291	110.46	67.67
NFTGraph-Tiny	20,000	245,221	1.30%	24.52	18,104	1,330	**738.27**	54.24
NFTGraph	1,161,847	2,851,407	0.39%	4.91	789,782	20,000	**160904.05**	4074.64

Definition of Extreme High-degree Node: Define a high-degree node in a graph as a node with a degree greater than the average degree (avg_deg). An extreme high-degree node is defined as a node whose degree/avg_deg$\geqslant q$ ($q\geqslant 1$), indicating that the node's degree exceeds the average degree by q times. q is a hyperparameter that varies depending on the dataset. Let q_1 denote the hyperparameter selectively elevating No. 1 node to an extreme high-degree node, while q_2 signifies the hyperparameter concurrently elevating both No. 1 and No. 2 to extreme high-degree nodes. As seen from Table 2, for NFTGraph, q_1 exceeds 16,0000, and q_2 also exceeds 4,000, demonstrating significant extreme high-degree node characteristics. NFTGraph-Tiny exhibits similar features. Furthermore, there is a considerable discrepancy between the values of q_1 and q_2, whereas for graph datasets including Weibo, Reddit, and Questions, q_1 and q_2 are closer, reflecting the significant extreme high-degree node characteristics of

No.1 node in NFTGraph and NFTGraph-Tiny. In the subsequent discussion, to better illustrate the characteristics of extreme high-degree nodes, unless otherwise specified, datasets are artificially restricted to feature only one extreme high-degree node designated by q_1. The extreme high-degree node is abbreviated as SN. (Visualization of SN of NFTGraph-Tiny is in the supplemental materials.)

4 Exploring the Influence of Extreme High-degree Node

4.1 Experimental Settings

Datasets: Due to the similarity properties between NFTGraph and NFTGraph-Tiny, and the challenges faced by certain GNNs in handling large graphs, the proposed NFTGraph-Tiny is chosen as the foundational dataset. To assess the influence of SN, a variant dataset is introduced by removing SN and the edges connected to it. These two graphs are respectively denoted as w/ SN and w/o SN. From Table 3, it can be observed that without SN, No.1 degree decreased from 18,104 to 1,330, bringing it closer to the degrees of its immediate neighbors. Moreover, to demonstrate the advantage of the proposed dataset, several commonly used and well-known graph anomaly datasets, namely Weibo [8,9], Reddit [9,20], and Questions [13], are selected for comparison. They have also undergone the operation of removing SN, as shown in Table 3.

Task Description: This section outlines a task aimed at identifying suspicious nodes. Formally, the objective is to train a model $f : f(u) \rightarrow \{0, 1\}$, where $\forall u \in \mathcal{V}$, $\mathcal{V}$ is node set, 1 denotes suspicious nodes and 0 denotes non-suspicious.

Models and Evaluation Metrics: To comprehensively evaluate the influnce of SN, this section selects 34 fraud detection models, including both supervised and unsupervised models, based on GNN and non-GNN models. Specifically, the unsupervised and non-GNN models [21] include OCSVM, LOF, CBLOF, COF, HBOS, SOD, COPOD, ECOD, LODA, and IForest; unsupervised and GNN-based models [9] include ANOMALOUS, ONE, OCGNN, CoLA, DONE, AnomalyDAE, CONAD, and DOMINANT; supervised and non-GNN models [16] include MLP, KNN, SVM, RF; supervised and GNN-based models [16] include GCN, SGC, GIN, GraphSAGE, GAT, GT, GAS, BernNet, AMNet, GHRN, GAT-Sep, PCGNN. Due to the severe class imbalance between suspicious and non-suspicious nodes, the Area Under the ROC Curve (AUROC) is chosen as the evaluation metric. Other settings can be found in the supplemental materials.

4.2 Overall Performance

Table 4 presents the AUROC of models on NFTGraph-Tiny, Weibo, Reddit, and Questions datasets, along with their corresponding graphs without SN . Overall, NFTGraph-Tiny poses a challenge for fraud detection, as all models exhibit AUROC below 0.7. One reason is its anomaly ratio is extremely low, and another is it comes from real fraud scenarios. Typically, anomalous nodes

Table 3: Datases for exploring the influence of SN

Datasets	#Nodes	#Edges	#Feature	#Anomaly	No.1-5 Deg	AnomalyAvgDeg
NFTGraph-Tiny w/ SN	20,000	245,221	50	259	[18104,1330,1212,1020,917]	27.66
NFTGraph-Tiny w/o SN	19,999	227,118	50	259	[1330,1211,1020,916,793]	27.66
Weibo w/ SN	8,405	416,368	400	868	[4447,2769,2723,2558,2523]	54.82
Weibo w/o SN	8,404	411,922	400	868	[2767,2721,2556,2521,2376]	54.82
Reddit w/ SN	10,984	168,016	64	366	[5112,3134,3106,2608,2518]	24.75
Reddit w/o SN	10,983	162,905	64	366	[3134,3106,2608,741,2476]	24.75
Questions w/ SN	48,921	202,461	301	1460	[1541,1186,901,741,739]	20.93
Questions w/o SN	48,920	200,921	301	1460	[1185,900,740,738,717]	20.93

are categorized into three types [9]: structural anomalies, attribute anomalies, and a combination of both. Structural anomalies refer to abnormal interactions of nodes with others and communities, while attribute anomalies pertain to abnormal attribute features of nodes themselves. However, the anomalous nodes in NFTGraph-Tiny do not adhere to this inherent classification. The average clustering coefficients of anomalous and non-anomalous nodes in this graph are 0.0682 and 0.0744, respectively, and the average neighbor feature similarities are 0.5327 and 0.5126, respectively. The former represents a structural anomaly indicator, while the latter represents an attribute anomaly indicator. It can be observed that in NFTGraph-Tiny, the values of both indicators for anomalous and non-anomalous nodes are very close, indicating the difficulty in detection due to the organic anomalies [9] from real scenarios. This highlights the advantage of the proposed graph anomaly dataset, particularly compared to injected or synthetic anomalies [9].

4.3 Influence of SN on GNN-based and non-GNN-based Models

Refining Table 4, the **significant change rate** is defined as the proportion of models with AUROC changes exceeding 2% (±2%) after removing SN, while the **positive significant change rate** indicates an augmentation in AUROC (+2%) after SN removal.

Table 5 delineates the significant change rate for non-GNN and GNN models. It is evident that the significant change rate for NFTGraph-Tiny surpasses that of Weibo, Reddit, and Questions. This discrepancy arises due to NFTGraph-Tiny possessing a notably higher q_1 of SN compared to the others. This underscores the advantage of NFTGraph-Tiny in elucidating the impact of SN, wherein the detection performance may significantly change before and after removing SN.

Across the four datasets, the significant change rates of GNN-based models are substantially higher than those of non-GNN-based models. Specifically, on the NFTGraph-Tiny dataset, the significant change rate for GNN-based models is 70.00%, whereas for non-GNN-based models, it is only 50.00%. In the cases of the Weibo and Questions datasets, the non-GNN-based models even experience no significant change in AUROC, while for Reddit, the significant change rate is also higher for GNN-based models. Therefore, the results indicate that the

Table 4: AUROC metrics of GAD. Bold for significant change of AUROC.

Models		Datasets	NFTGraph-Tiny		Weibo		Reddit		Questions	
			w/ SN	w/o SN	w/ SN	w/o SN	w/ SN	w/o SN	w/ SN	w/o SN
Unsupervised non-GNN-based	OCSVM		0.4763	**0.5018**	0.8001	0.8017	0.5702	0.5703	0.5995	0.5995
	LOF		**0.5658**	0.5332	0.5756	0.5756	0.5369	0.5372	0.5680	0.5679
	CBLOF		0.5134	0.5106	0.8003	0.8084	0.5809	0.5827	0.6016	0.6003
	COF		**0.5662**	0.5430	0.4877	0.4885	0.5755	0.5756	0.5591	0.5591
	HBOS		0.4998	0.5041	0.4038	0.4034	0.5338	0.5338	0.5951	0.5951
	SOD		0.6590	0.6405	0.4258	0.4249	0.5495	0.5402	0.5526	0.5553
	COPOD		0.5977	0.5977	0.4736	0.4738	0.4974	0.4975	0.6059	0.6059
	ECOD		0.5240	0.5241	0.4774	0.4775	0.4999	0.4999	0.6015	0.6015
	LODA		**0.5749**	0.5412	0.7139	0.7096	0.5630	0.5633	0.5745	0.5745
	IForest		0.6016	0.6009	0.5500	0.5502	**0.5942**	0.5514	0.6057	0.6020
Unsupervised GNN-based	ANOMALOUS		0.6159	**0.6818**	0.9876	0.9876	0.5688	0.5629	0.5527	0.5530
	ONE		**0.5445**	0.4992	0.6637	0.6518	**0.5356**	0.5157	0.4867	**0.5102**
	OCGNN		**0.6327**	0.5389	0.8251	0.8257	**0.6308**	0.6139	0.5590	**0.5745**
	CoLA		**0.4943**	0.4021	0.4254	**0.4464**	0.4963	**0.5409**	0.5306	0.5465
	DONE		0.5734	**0.5858**	0.5536	**0.6569**	0.5518	0.5556	0.6644	0.6639
	AnomalyDAE		0.5555	**0.5803**	0.8256	0.8268	0.5805	0.5709	0.4771	**0.4995**
	CONAD		0.5382	0.5424	0.6311	**0.7050**	0.4680	**0.5174**	0.6019	0.6021
	DOMINANT		0.6026	**0.6251**	**0.7015**	0.6290	0.5129	0.5138	0.6036	0.6028
Supervised non-GNN-based	MLP		0.5645	**0.6730**	0.9738	0.9669	0.6771	0.6765	0.6753	0.6785
	KNN		0.5994	**0.6204**	0.9672	0.9674	0.6067	**0.6301**	0.6760	0.6789
	SVM		0.5756	0.5773	0.9536	0.9539	0.6622	0.6659	0.6359	0.641
	RF		**0.6539**	0.6314	0.9864	0.9865	0.6290	0.6312	0.5621	0.5512
Supervised GNN-based	GCN		0.6580	0.6401	0.9830	0.9867	0.7172	0.7122	0.7018	0.7011
	SGC		0.5968	**0.6179**	0.9892	0.9893	0.6842	0.6885	0.6911	0.6921
	GIN		**0.6688**	0.6164	0.9881	0.9901	**0.7028**	0.6574	0.7185	0.7185
	GraphSAGE		0.5777	**0.6437**	0.9934	0.9932	0.6949	**0.7130**	0.7197	0.7179
	GAT		0.6510	0.6405	0.9800	0.9816	**0.6866**	0.6724	0.7037	0.7093
	GT		0.6163	**0.6518**	0.9899	0.9897	0.6444	**0.6682**	0.6949	0.7134
	GAS		**0.6663**	0.6636	0.9828	0.9824	**0.6858**	0.6627	**0.7118**	0.6913
	BernNet		0.6230	**0.6628**	0.9783	0.9853	**0.6868**	0.6763	0.6951	0.7095
	AMNet		**0.6970**	0.6601	0.9808	0.9858	0.6445	0.6371	0.699	0.6989
	GHRN		0.6734	0.6656	0.9792	**0.9892**	0.6894	**0.7180**	0.7204	0.7210
	GAT-Sep		**0.6775**	0.6534	0.9846	0.9863	0.6665	0.6739	0.6913	0.6892
	PCGNN		**0.6895**	0.6377	0.9848	0.9846	0.6779	0.6785	**0.6929**	0.6692

impact of SN on GNN-based detection models is greater than that on non-GNN-based detection models.

Moreover, Table 5 reveals that the positive significant change rate for GNN-based models without SN exceeds 50%. This phenomenon contradicts intuition, as the informational content with SN is presumed to be no less than that without SN. Consequently, one would expect the AUROC of the graph with SN to consistently outperform that without SN. However, more than half of the GNNs exhibit higher AUROC values after SN removal, with certain GNNs even witnessing a notable increase of up to 7 percentage points in AUROC (e.g., ANOMALOUS). This phenomenon suggests that SN and its associated edges may introduce noise, which, through neighbor aggregation, complicates the representation modeling of nodes, rendering the differentiation between suspicious and non-suspicious

Table 5: Significant Change Rate for Non-GNN and GNN Models

	NFTGraph-Tiny	Weibo	Reddit	Questions
non-GNN-based	**50.00%**	0.00%	14.29%	0.00%
GNN-based	**70.00%**	20.00%	40.00%	25.00%
non-GNN-based w/o SN +	42.86%	-	50.00%	-
GNN-based w/o SN +	50.00%	75.00%	50.00%	60.00%

nodes more difficult. This insight highlights the importance of considering the potential noise introduced by SN for GNNs, due to its numerous edges.

4.4 Impact of SN on Unsupervised and Supervised GNNs

Given that GNN-based models generally exhibit higher significant change rates compared to non-GNN-based models, the experimental results of GNN-based models are further analyzed to assess the influence of SN on unsupervised and supervised settings.

From Table 6, it can be observed that, regardless of the supervised or unsupervised setting, the significant change rate of NFTGraph-Tiny is not lower than that of the other three datasets. Specifically, the significant change rate of supervised GNN models on NFTGraph-Tiny reaches 66.67%. In contrast, the significant change rates for Reddit and Questions are 33.33% and 16.67%, respectively, while that for Weibo is 0, primarily due to the extreme high-degree SN of NFTGraph-Tiny.

Table 6: Significant Change Rate for Unsupervised and Supervised GNN Models

	NFTGraph-Tiny	Weibo	Reddit	Questions
Unsupervised GNN	**50.00%**	50.00%	50.00%	37.50%
Supervised GNN	**66.67%**	0.00%	33.33%	16.67%
Unsupervised GNN w/o SN +	75.00%	75.00%	50.00%	100.00%
Supervised GNN w/o SN +	50.00%	-	50.00%	0.00%

Additionally, across the four datasets, the positive significant change rates of unsupervised GNN models are higher than those of supervised GNN models. This suggests that, after removing SN, unsupervised GNN models achieve a higher proportion of models with increased AUROC. This phenomenon may be attributed to the absence of training labels in unsupervised GNN models, making the noise introduced by SN edges more impactful for fraud detection. Consequently, after removing SN, unsupervised GNN models may learn better.

4.5 Computational Cost

Given the advantages of NFTGraph-Tiny highlighted above, this section through Sec. 4.7 utilizes NFTGraph-Tiny for investigation. Moreover, to better illustrate the influence on computational cost, NFTGraph is also utilized in this section.

On the datasets, we present statistics regarding average number of node neighbors and execution time of both 1-layer and 2-layer GAT. From Table 7, it can be observed that the average number of 1-hop neighbors of GAT in NFTGraph-Tiny is 11.26, while the average number of 2-hop neighbors reaches 2386.39. Upon removing SN, the average number of 1-hop neighbors remains relatively unchanged, while the average number of 2-hop neighbors sharply decreases to 123.88. This is due to the fact that the degree of SN in NFTGraph-Tiny is 18,104, indicating that the majority of the whole 20,000 nodes in the graph are connected to SN, resulting in their 2-hop neighbors including the 1-hop neighbors of SN. More pronounced disparities are observed in NFTGraph and NFTGraph w/o SN: the difference in average 2-hop neighbors before and after removing SN is significant, with the former being over 1300 times the latter.

Table 7: Average number of node neighbors for GAT at different hops

Dataset / Hops	1-hop	2-hop
NFTGraph-Tiny	11.26	**2386.39**
NFTGraph-Tiny w/o SN	10.36	123.88
NFTGraph	2.45	**27647.54**
NFTGraph w/o SN	1.86	20.42

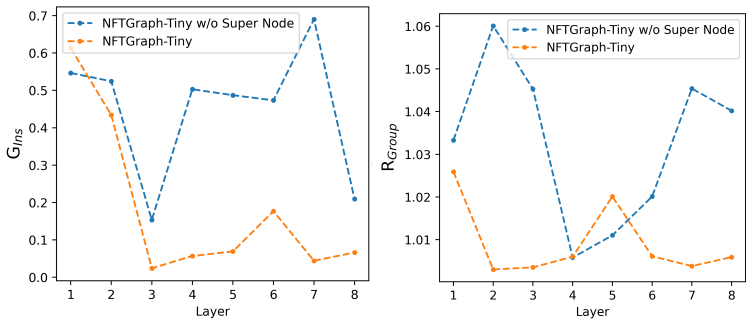
Table 8: Execution Time (s) of GAT with different numbers of layers

Dataset / Layer Number	1-layer	2-layer
NFTGraph-Tiny	4.89	**5.84**
NFTGraph-Tiny w/o SN	4.58	5.31
NFTGraph	49.26	**91.0**6
NFTGraph w/o SN	20.03	33.73

Table 8 demonstrates how the execution time of GAT varies with different numbers of layers. Notably, on NFTGraph-Tiny, removing SN leads to a non-linear decrease in execution time, with a more significant reduction observed for 2-layer GAT compared to a single layer. Similarly, in NFTGraph, the execution time for a 2-layer GAT is nearly one-third of the original time after SN removal, resulting in a considerable reduction. Therefore, the presence of SN significantly impacts the computational cost, leading to a substantial increase in both the average number of node neighbors and the execution time.

4.6 Over-smoothing

To investigate the influence of extreme high-degree nodes on over-smoothing, we compute two over-smoothing metrics [22]: Instance Information Gain (G_{Ins}) and Group Distance Ratio (R_{Group}). These calculations are performed across different layer numbers of GAT applied to both the NFTGraph-Tiny dataset and its variant (w/o SN). Generally, lower values of these metrics indicate a higher level of over-smoothing.



(a) Over-smoothing metric G_{Ins} (b) Over-smoothing R_{Group}

Fig. 2: Over-smoothing metrics G_{Ins} and R_{Group} for different layer numbers of GAT on NFTGraph-Tiny and NFTGraph-Tiny w/o SN

Figure 2 illustrates the changes in the over-smoothing metrics G_{Ins} and R_{Group} for different layer numbers of GAT. It is evident that, in the majority of cases, as the number GAT's layers increases, the metrics decrease, indicating a progressive over-smoothing of node representations. Upon removal of SN, both G_{Ins} and R_{Group} metrics exhibit an increase compared to the original, thereby alleviating the over-smoothing phenomenon. Therefore, the results demonstrate that with the same layer number of GNN, the presence of extreme high-degree nodes increases the likelihood of over-smoothing.

4.7 Exploration Under Different Thresholds

To explore the influence of extreme high-degree nodes under different thresholds q, experiments are conducted to evaluate the AUROC results of GAT on head and tail nodes of the degree distribution. To be more illustrative, here the threshold q corresponds to the number of head nodes. For instance, if the threshold is set to 10%, it means that a suitable q value is selected so that the top 10% nodes in terms of degree in the graph are considered as extreme high-degree nodes.

Figure 3 shows that when the threshold is set at 10%, the model's prediction results for the top 10% head nodes are greater than those for the tail 10% nodes. As the threshold relaxes, both the head and tail nodes' prediction AUROC decreases, but the prediction results for head nodes remain greater than those for tail nodes. The reason is that head nodes have higher degrees and more connections with neighbors, which enables the model to better capture node representations and facilitates detection. Therefore, nodes with higher degrees, especially extreme high-degree nodes, are more likely to be accurately detected, while suspicious nodes, to avoid exposure, are often involved in limited interactions, increasing the difficulty of modeling and detection. Table **??** also presents the average degree of anomalies.

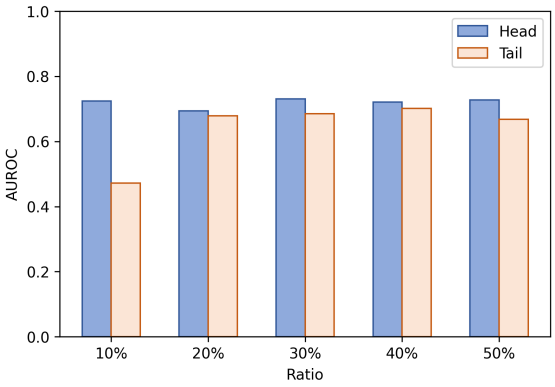


Fig. 3: AUROC for head and tail Nodes at the same proportion

5 Method and Experiments

5.1 SNGNN

To mitigate the influence of extreme high-degree nodes, we propose a graph fraud detection algorithm termed Super Node-Aware Graph Neural Network (SNGNN). The schematic diagram of SNGNN is shown in Figure 4. The design of SNGNN considers the conclusions above. Overall, these two modules mainly address two main issues: the problem of low-degree fraud nodes and the noise problem in SN edges. The main components of SNGNN consist of two modules:

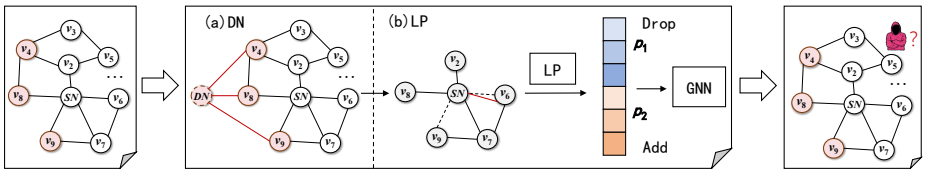


Fig. 4: SNGNN

Module (a): Introducing a dummy node (DN). This node serves as a virtual fraud node and is connected to all fraud nodes in the graph. This design helps alleviate the low degrees of fraudulent nodes to a certain extent (this module is denoted as DN).

Module (b): Link prediction for SN. This module takes SN and their neighbors as input into a link predictor (LP). The LP outputs the probabilistic existence of edges between SN and their neighbors. Then, based on the output vector, the p_1 and p_2 percentiles are extracted. Nodes with probabilities less than p_1 are disconnected from SN, while nodes with probabilities greater than p_2 are connected to SN. Subsequently, input the new graph into a GNN to obtain

the representations of nodes. This design aids in reducing noise within SN edges (this module is denoted as LP).

5.2 Experiments

Setup: NFTGraph-Tiny, Weibo, Reddit, and Questions datasets are used, and the task is anomaly detection. The baseline models include three basic GNNs: GCN, GAT, and GraphSAGE, as well as three GNNs that achieve better performance in Table 4: PCGNN, GAS, and GIN. Settings are different from Sec.4.1, which can be found in the supplemental materials.

Table 9: Comparison of AUROC between SNGNN and other GADs

Model / Dataset	NFTGraph-Tiny	Weibo	Reddit	Questions
GCN	0.5953	0.9875	0.7189	0.6819
GAT	0.6226	0.9902	0.6733	0.7167
GraphSAGE	0.6428	0.9917	0.6800	0.7259
PCGNN	0.5832	0.9848	0.7079	0.6784
GAS	0.5552	0.9915	0.6996	0.7111
GIN	0.5929	0.9908	0.6872	0.7160
SNGNN	**0.6980**	**0.9926**	**0.7272**	**0.7325**

Results: Table 9 shows the comparison of AUROC between SNGNN and other GADs. On all four datasets, SNGNN achieves the highest AUROC. On the NFTGraph-Tiny dataset, SNGNN achieves an AUROC of 0.6980, surpassing GraphSAGE by 5%. On the Weibo dataset, SNGNN's AUROC is 0.9926, which is approximately 1% higher than GraphSAGE. Similarly, on Reddit and Questions, SNGNN outperforms GCN (0.7189) and GraphSAGE (0.7259) models, bringing about a 1% AUROC improvement. It is worth noting that SNGNN achieves at least a 5% AUROC improvement on NFTGraph-Tiny, significantly higher than the approximately 1% improvement on other datasets. The disparity in performance is due to the notably higher degree values of SN in NFTGraph-Tiny compared to Weibo, Reddit, and Questions, as shown in Table 3, highlighting SNGNN's superior performance in this context.

5.3 Ablation Study

To validate the effectiveness of each module in SNGNN, we design several ablation tests. Specifically, while keeping the other parts and hyperparameters unchanged, the DN module (referred to as w/o DN) and the LP module (referred to as w/o LP) are removed separately, and then the performance is observed.

Table 10 shows the results of ablation tests. It can be observed that after removing the DN module, the detection AUROC of SNGNN decreases. Especially on the NFTGraph-Tiny and Questions datasets, the AUROC of SNGNN

Table 10: Ablation study results for SNGNN

Model / Dataset	NFTGraph-Tiny	Weibo	Reddit	Questions
SNGNN w/o DN	0.6414	0.9914	0.7188	0.7086
SNGNN w/o LP	0.6351	0.9919	0.7239	0.7270
SNGNN	**0.6980**	**0.9926**	**0.7272**	**0.7325**

decreases by more than 5% and 3%, respectively. These results validate the effectiveness of DN, indicating that by introducing the dummy node, it can alleviate the problem of low degree of fraudulent nodes and enhance the information propagation among fraudulent nodes. Similarly, removing the LP module also leads to a decrease in AUROC of SNGNN, with a decrease of over 6% observed on the NFTGraph-Tiny dataset. Therefore, LP also plays an important role in SNGNN. The introduction of LP can eliminate the noise in the edges of SN, and the threshold setting ensures the confidence of denoising.

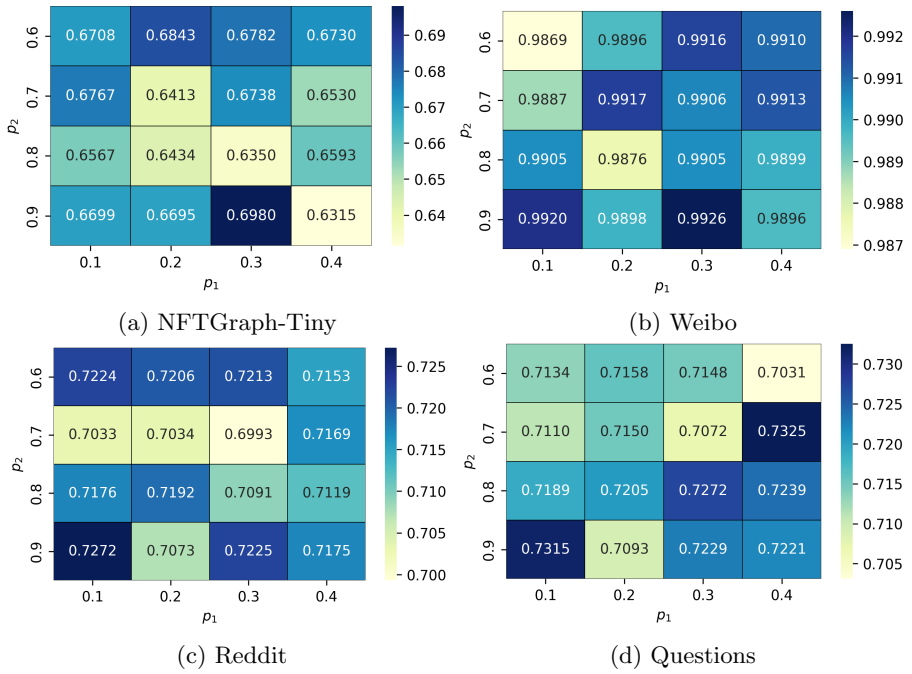
5.4 Parameter Sensitivity Analysis

The hyperparameters p_1 and p_2 represent the p_1 and p_2 percentiles of the LP output probabilities, determining the removal and addition of edges of SN in each training iteration. p_1 and p_2 indicate the strength and confidence of edge removal (addition), thus it is necessary to analyze the sensitivity to p_1 and p_2.

Figure 5 shows the detection AUROC of the SNGNN model under different values of p_1 and p_2 on four datasets. The optimal values of p_1 and p_2 for NFTGraph-Tiny and Weibo are both (0.3, 0.9), for Reddit is (0.1, 0.9), and for Questions is (0.4, 0.7). The optimal values for these four datasets are not located in the central area (i.e., $0.2 \leqslant p_1 \leqslant 0.3$, $0.7 \leqslant p_2 \leqslant 0.8$), indicating that for the edges of SN, the best practice is to ensure that the number of edge removals and additions is either relatively high or relatively low, or one is high while the other is low. If the number of edge removals and additions in each iteration is kept at a moderate level, it is difficult to achieve the optimal state.

6 Conclusion

In this paper, we aim to investigate the influence of extreme high-degree nodes (SN) on GAD and to mitigate this influence. To achieve this goal, we introduce a new graph dataset and analyze SN's influence including whether based on GNN, supervised or unsupervised settings, computational costs, etc. To eliminate the influence, we propose SNGNN, which significantly outperforms existing models with an average detection AUROC improvement of over 2%. SNGNN demonstrates effectiveness in detecting anomalies in graphs with extreme high-degree nodes. Our future work aims to further enhance our algorithm to eliminate SN's influence on computational costs and over-smoothing, thereby achieving even higher detection AUROC.



(a) NFTGraph-Tiny (b) Weibo

(c) Reddit (d) Questions

Fig. 5: Parameter sensitivity

References

1. Bandyopadhyay, S., Vivek, S.V., Murty, M.: Outlier resistant unsupervised deep architectures for attributed network embedding. In: Proceedings of the 13th international conference on web search and data mining. pp. 25–33 (2020)
2. Chen, L., Peng, J., Liu, Y., Li, J., Xie, F., Zheng, Z.: Phishing scams detection in ethereum transaction network. ACM transactions on internet technology(TOIT) **21**(1), 1–16 (2020)
3. Chen, W., Zheng, Z., Cui, J., Ngai, E., Zheng, P., Zhou, Y.: Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In: Proceedings of the world wide web conference(WWW). pp. 1409–1418 (2018)
4. Ding, K., Li, J., Bhanushali, R., et al.: Deep anomaly detection on attributed networks. In: Proceedings of the international conference on data mining(SDM). pp. 594–602 (2019)
5. Giles, C.L., Bollacker, K.D., Lawrence, S.: Citeseer: An automatic citation indexing system. In: Proceedings of the third conference on digital libraries. pp. 89–98 (1998)
6. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in Neural Information Processing Systems(NeurIPS) **30** (2017)
7. Huang, Y., Wang, L., Zhang, F., Lin, X.: Unsupervised graph outlier detection: Problem revisit, new insight, and superior method. In: Proceedings of 39th international conference on data engineering(ICDE). pp. 2565–2578 (2023)
8. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 27th sigkdd conference on knowledge discovery and data mining(SIGKDD). pp. 1269–1278 (2019)

9. Liu, K., Dou, Y., Zhao, Y., Ding, X., Hu, X., Zhang, R., Ding, K., Chen, C., Peng, H., Shu, K., et al.: Bond: Benchmarking unsupervised outlier node detection on static attributed graphs. Advances in Neural Information Processing Systems(NeurIPS) **35**, 27021–27035 (2022)
10. Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., He, Q.: Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In: Proceedings of the world wide web conference(WWW). pp. 3168–3177 (2021)
11. LIU, Z., NGUYEN, T.K., FANG, Y.: On generalized degree fairness in graph neural networks. In: Proceedings of the 37th international conference on artificial intelligence(AAAI). pp. 7–14 (2023)
12. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: Proceedings of the 22nd international conference on blockchain. pp. 897–908 (2013)
13. Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., Prokhorenkova, L.: A critical look at the evaluation of gnns under heterophily: are we really making progress? arXiv preprint arXiv:2302.11640 (2023)
14. Rayana, S., Akoglu, L.: Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th sigkdd conference on knowledge discovery and data mining(SIGKDD). pp. 985–994 (2015)
15. Tang, J., Hua, F., Gao, Z., Zhao, P., Li, J.: Gadbench: Revisiting and benchmarking supervised graph anomaly detection. Advances in Neural Information Processing Systems(NeurIPS) **36** (2024)
16. Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: Proceedings of the international conference on machine learning(ICML). pp. 21076–21089 (2022)
17. Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., Mitra, P., Wang, S.: Investigating and mitigating degree-related biases in graph convoltuional networks. In: Proceedings of the 29th international conference on information and knowledge management(CIKM). pp. 1435–1444 (2020)
18. Wu, J., He, J., Xu, J.: Demo-net: Degree-specific graph neural networks for node and graph classification. In: Proceedings of the 25th sigkdd conference on knowledge discovery and data mining(SIGKDD). pp. 406–415 (2019)
19. Xu, Z., Huang, X., Zhao, Y., Dong, Y., Li, J.: Contrastive attributed network anomaly detection with data augmentation. In: Advances in knowledge discovery and data mining: 26th pacific-asia Conference (PAKDD). pp. 444–457. Springer (2022)
20. Zhao, T., Deng, C., Yu, K., Jiang, T., Wang, D., Jiang, M.: Error-bounded graph anomaly loss for gnns. In: Proceedings of the 28th international conference on information and knowledge management(CIKM). pp. 1873–1882 (2020)
21. Zhao, Y., Nasrullah, Z., Li, Z.: Pyod: A python toolbox for scalable outlier detection. Journal of machine learning research **20**(96), 1–7 (2019)
22. Zhou, K., Huang, X., Li, Y., Zha, D., Chen, R., Hu, X.: Towards deeper graph neural networks with differentiable group normalization. Advances in Neural Information Processing Systems(NeurIPS) **33**, 4917–4928 (2020)