Query-Based Model Extraction Attack on GCN: A Surrogate Model Technique for Non-Euclidean Data

Sibtain Syed[1], Alvi Ataur Khalil[2], Kishor Datta Gupta[3], Saima Jabeen[4], Mohammad Ashiqur Rahman[5]

[1]COMSATS University Islamabad, Pakistan
[2]Computer Science, Southern Illinois University Carbondale, USA
[3]Department of Cyber Physical System, Clark Atlanta University, USA
[4]College of IT and Business, Riyadh Elm University, KSA
[5]Analytics for Cyber Defense (ACyD) Lab, Florida International University, USA
sibtainshah621@gmail.com, alviataur.khalil@siu.edu, kgupta@cau.edu, saima.jabeen@riyadh.edu.sa, marahman@fiu.edu

Abstract

Machine learning (ML) models are facing serious threats from Model Extraction Attacks, in which a black-box model owned by a private service provider can be cloned to a surrogate model by an attacker pretending to be a client solely through query-based access. Unfortunately, most of the past studies only focus on ML models, which are trained on Euclidean spaces like images and texts, while model extraction attacks on Graph Neural Network (GNN) models containing node features and graph structure need to be explored. The respective study focuses on investigating and developing a model extraction attack strategy against a Graph Convolutional Network (GCN) model by simulating more realistic conditions for the attacker. The study begins by formalizing threat modeling based on GCN extraction attacks, categorizing potential threats in accordance with the levels of background knowledge accessible to the attacker, such as node attributes and neighbor connections. Subsequently, the study presents a novel method that leverages a learnable feature synthesis module in order to infer missing attributes of unknown neighbor nodes, evaluated using fidelity (85-90 percentage) and KL-divergence (0.28-0.10) to assess behavioral similarity with the victim model, rather than exact parameter recovery. Results demonstrate that even with partial knowledge, the majority of inputs in the target domain yield predictions identical to the original model.

Introduction

Euclidean data has been used in modeling various scenarios—such as healthcare, finance, optical character recognition (OCR), and water resource management—to foster more efficient outcomes (Syed et al. 2024a,c; Ahmed, Bibi, and Syed 2023; Syed et al. 2024b, 2023). Non-Euclidean data is equally important, supporting applications like social networks, rating systems, and document collections (Jagielski et al.; Gasteiger, Bojchevski, and Günnemann 2018). To analyze such graphs, GNNs have emerged with state-of-the-art performance (Zhu et al. 2020). Since deep learning models require costly data and computation, they are considered intellectual property (IP) (Hong et al. 2023). Model extraction (or stealing) replicates a black-box model's functionality using only query access (Figure 1), enabling adversaries to exploit the surrogate for financial or malicious purposes. Unlike adversarial attacks that degrade performance, extraction reconstructs an alternative model, often matching the victim's accuracy. Prior work has mainly addressed Euclidean data models (MLPs, CNNs), with limited focus on graph-structured data. We develop an attack strategy to extract a GCN. In this scenario, a private GCN is deployed on a server; the attacker queries selected nodes and trains a surrogate using responses. Unlike standard NN attacks, GNN extraction is harder since node classification depends on attributes, neighbors, and connectivity. Thus, attackers must cope with incomplete graph information, as in social networks where edges may be visible but features hidden. These constraints make GNN model extraction under partial knowledge technically challenging. To address this challenge, we propose a framework that simulates real-world attacker capabilities across three knowledge dimensions: attack node attributes, partial graph structures, and auxiliary subgraphs with similar properties. The core contribution lies in a model extraction strategy that leverages a learnable feature synthesizer to infer hidden attributes of unreachable neighbors, while also formalizing threat modeling from the perspectives of structure, attributes, and shadow subgraphs. In addition, we present an evaluation approach that combines traditional accuracy with KL divergence, thereby measuring both label-level replication and probabilistic similarity between the victim and surrogate models. This dual view provides a deeper understanding of model fidelity. Finally, experiments on three real-world datasets confirm that our surrogate models replicate victim performance with near-identical accuracy and almost 90% prediction similarity. Although the attack itself operates only with hard-label outputs, we also report KL divergence in a separate soft-label evaluation (assuming logits are observable) to quantify distributional alignment; importantly, this metric is never accessible to the attacker.

Related Work

Model extraction (model stealing) replicates a black-box model via query access. For linear models, prediction-API attacks are well established (Tramèr et al. 2016),

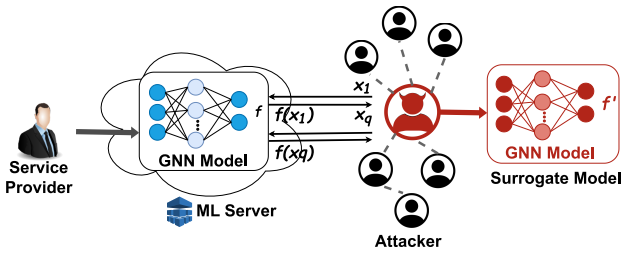

Figure 1: An Architecture of Model Extraction Attacks on GNN. A service provider owns a server that deploys a GNN model f. Attacker extracts a surrogate model $f' \approx f$ based on the values returned from the server.

and reinforcement learning can select informative queries (Orekondy, Schiele, and Fritz 2019). For deep networks, gradient-based extraction of two-layer ReLU models (Milli et al. 2019) and active-learning attacks on text/image classifiers (Pal et al. 2019) have been demonstrated. Defenses that hide or perturb probabilities, or detect malicious queries via entropy/input-distribution analysis, are limited—ineffective against label-only extraction and poorly suited to GNNs (Chandrasekaran et al. 2018; Juuti et al. 2018). Most GNN work studies adversarial robustness rather than extraction (Li et al. 2020b): greedy perturbation analyses (Zügner, Akbarnejad, and Günnemann 2018), transferable attacks (Zhang et al. 2021a), fake-node injection (Wang and Gong 2019), backdoors (Zhang et al. 2021b; Xu, Xue, and Picek 2021), and community obfuscation via surrogates (Li et al. 2020a). Confidentiality threats such as membership inference (Gong and Liu 2018; Wu et al. 2021a) and link stealing (He et al.) target components of the graph rather than duplicating the full model. Our work addresses this gap by directly tackling GNN model extraction and its broader implications. Prior studies have also considered heuristics technique for neighboring node predictions (Wu et al. 2021b), in reference to feature of neighboring node are highly correlated (Wang et al. 2020; Acharya and Zhang 2024). However, we investigated a neural network base feature synthesizer for attribute predictions rather than totally relying on symbolic approach.

Preliminaries

Model Extraction Attacks: Given a target classifier $f_\theta(x)$ with parameters θ_T, an attacker queries the model with unlabeled data to obtain responses $p_i = f_\theta(p)$. Using these, a surrogate model $f_{\theta'}(x)$ with parameters θ_S is trained to approximate the target. In the *soft-label* case, probability vectors are returned, whereas the *hard-label* case only yields the top-1 prediction.

Node Classification: Consider an attributed graph $Z = (p, E, F)$, where p denotes nodes, E edges, and F node features. A black-box node classifier $f_\theta(x)$ assigns labels Y to nodes by leveraging both attributes and structural context. For a node $p_i \in p$, the predicted output is $p_i = f_\theta(p_i)$, which ideally aligns with the ground-truth label y_i.

Graph Convolutional Networks: We adopt a two-layer GCN (Kipf and Welling 2016) for node classification, denoted $f_\theta(x)$. A GCN aggregates features from neighbors and is defined as $f_\theta(A, F) = softmax\big(\hat{A} \cdot ReLU(\hat{A} \cdot F \cdot W^{(0)}) \cdot W^{(1)}\big)$, where $\hat{A} = \hat{D}^{-1/2}\tilde{A}\hat{D}^{-1/2}$ is the normalized adjacency, $\tilde{A} = A + I_n$ adds self-loops, and $\hat{D}_{ii} = \sum_j \tilde{A}_{ij}$. $W^{(0)}$ and $W^{(1)}$ are trainable weights, with $ReLU(a) = \max(0, a)$. Table 1 summarizes the notations. For extraction attacks on GCNs, the goal is to reconstruct $W = \{W^{(0)}, W^{(1)}\}$.

Symbol	Description	Symbol	Description
Z	Victim graph	Z'	Cloned graph
p	Nodes of Z	P_A	Attack nodes
Y	Node labels	$p_{A,k}$	k-hop neighbors of P_A
F	Node features	$E_{A,k}$	k-hop neighbor edges
E	Edges of Z	$F^*_{A,k}$	Synthetic neighbor features
$f_\theta(x)$	Node classifier	E_A	Edges among P_A
P	Predictions of f_θ	F_A	Features of P_A
D_i	Degree of p_i	E^*_A	Synthetic edges of P_A

Table 1: Summary of Symbols

Attack Detail

We consider a cloud-deployed GNN with a query interface where an attacker submits inputs and receives outputs to train a surrogate. Specifically, a GCN $f_\theta(x)$ trained on graph Z is targeted, and the adversary seeks to build $f_{\theta'}(x)$ such that $\forall p_i \in p,\ f_{\theta'}(p_i) \approx f_\theta(p_i)$. An attack is successful when the surrogate achieves comparable accuracy to the target, even if weights or architecture differ. Thus, privacy is compromised if decision behavior is replicated, not necessarily the exact parameters.

Adversary's Knowledge: We focus on black-box attacks, where adversaries may query outputs but lack access to hyperparameters, labels, or probabilities. They may only know partial node features, limited subgraph structure, or shadow data. In practice, attackers often see only their own nodes and edges (e.g., in social or financial networks).

Node Features F: Attackers may obtain features for attack nodes P_A (e.g., user profiles) but not for all nodes, as many attributes remain hidden.

Graph Structure A: Knowledge of edges linked to P_A allows subgraph reconstruction. While features may be private, relationships like friendships can be public, enabling crawling or inference of structure (He et al. 2020).

Shadow Data Z′: Attackers may access subgraphs from the same domain (e.g., different communities in a large network) to support surrogate training (He et al. 2020).

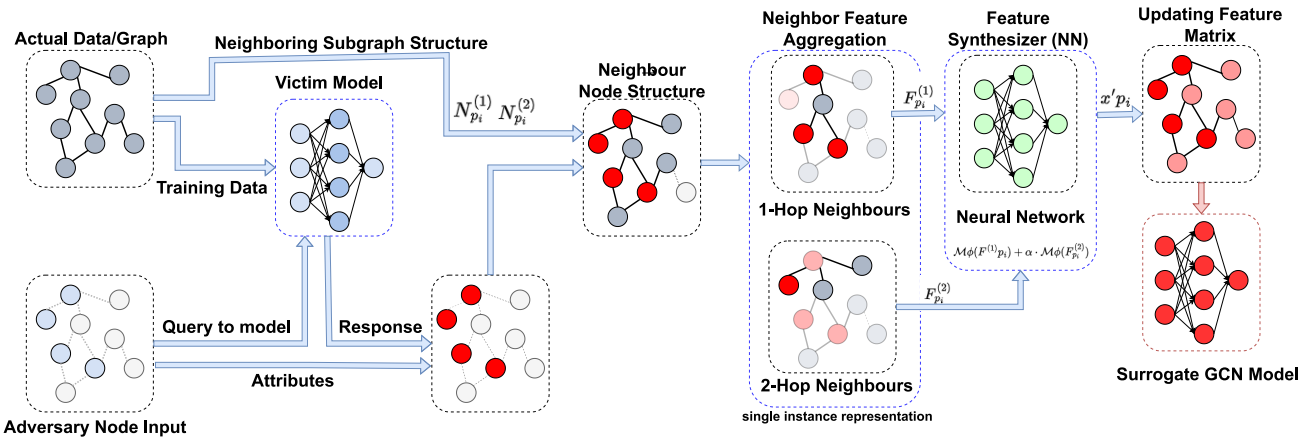

Figure 2: Architecture of the proposed GNN extraction attack.

Attack Classification: We assume attackers know partial F and E (features and edges) but lack shadow data. This reflects realistic cases where adversaries control subsets of nodes (e.g., multiple accounts in a financial network), seeing their features and connections. Full access to all nodes or edges represents an unrealistic adversarial assumption.

Proposed Attack Model

The study considers an attack scenario in which an attacker is able to acquire P_A and has access to F_A and neighbor sub-graph $A_{A,k-\text{hop}}$. These P_A are randomly selected from p to simulate a real world case in which any of the nodes in the original graph can be selected as P_A and thus make them potential attack nodes. To replicate the target model, an attacker tries to construct a graph specifically for the training surrogate model, called the *attack graph*. This attack graph covers F_A, E_A, and the node labels. An attacker utilizes adversarial knowledge to acquire (or to synthesis) these components. In particular, construction of attack graph for training a surrogate model mainly consists of three distinct steps to assemble these elements. Figure 2 illustrates the process of constructing the attack graph as discussed.

Issue Queries and Request Labels Based on our assumptions, an attacker can obtain the attack node attributes and response responses. In this case, nodes responses to queries are considered their node labels. Therefore, they are used to train a surrogate node classification model.

Collecting Edges of Neighbor Nodes By leveraging the input attributes, connections between the attack nodes, and predictions, an attacker could effectively train a surrogate model using supervised learning approach. However, neighboring nodes of attack nodes can influence their output predictions. Training node classification model using just attack nodes in isolation from the rest of the graph disregards their neighboring node influencing ability, leading to reduced performance. Consequently, it is essential to account for attributes of neighboring nodes surrounding the attack nodes. To address this, an attacker collects the edges of attack nodes along with their neighbors, which collectively represents the attack graph's structure.

Synthesizing Node Attributes for Unreachable Nodes The study assumes that the attacker has access only to the attributes and query responses of attack nodes. Consequently, an attacker must generate synthetic attributes of neighboring nodes with hidden attributes. Empirical observations indicate that the majority of nodes shares similar features with their neighboring nodes. Utilizing this insight, synthetic attributes could be formulated as a combination of attributes of their neighboring nodes. yet, to overcome this rule-based synthesis we employ a neural feature synthesizer, a shallow feedforward network trained on accessible node features to generate attributes for neighboring nodes with hidden features. For each synthetic node, 1-hop and 2-hop neighbors are identified, their normalized features collected, and passed to the synthesizer $\mathcal{M}_\phi$ to generate learned representations in equation 1.

$$x'p_i = \mathcal{M}\phi(F^{(1)}p_i) + \alpha \cdot \mathcal{M}\phi(F^{(2)}_{p_i}) \tag{1}$$

This learned synthesis replaces prior heuristic averaging, and adapts to feature distributions present in the graph domain.

Training the Feature Synthesizer M_ϕ We instantiate M_ϕ as a two-layer MLP (hidden size 64, ReLU, layer norm). To enable use under partial visibility, we pretrain M_ϕ only on neighborhoods whose features are visible to the attacker (i.e., nodes in P_A whose 1–2-hop neighbors also lie in P_A. For each such node P_i, we construct inputs $F_i^{(1)}, F_i^{(2)}$(normalized 1- and 2-hop neighbor feature aggregates), and the target x_i. We minimize μ_{synth} as shown in equation 2:

$$\mu_{synth} = |M_\phi(F_i^{(1)}) + \alpha M_\phi(F_i^{(2)}) - x_i|_2^2 \tag{2}$$

then freeze M_ϕ and use it to synthesize features for neighbors with hidden attributes, as shown in equation 1. This preserves the hard-label attack surface while improving the realism of synthesized neighborhoods."

Extracted Model Learning Once the features of these nodes are generated, an attacker constructs a Z'e that encompasses P_A along with neighboring nodes ($p_{A,k\text{-}hop}$) by incorporating both $F^*_{A,k\text{-}hop}$ and F_A. This graph is then used to train a GCN surrogate node classification model. It is important to note that the synthetic nodes remain unlabeled, as their labels cannot be directly obtained by attacker. Unlike the attack nodes, whose labels are derived from responses from the query, synthesis nodes are obscured from attacker, preventing any modification of their attributes or querying of the target models. Consequently, only P_A are labeled and the surrogate model is trained using a semi-supervised learning approach. The proposed attack strategy is represented in Algorithm 1.

Algorithm 1: Learnable Feature-Based Model Extraction

Input: F_A: attributes of attack nodes; P_A: query responses; α: adjustment factor; $(P_{A,2\text{-hop}}, E_{A,2\text{-hop}})$: 2-hop subgraph; $\mathcal{M}_\phi$: feature synthesizer.
Output: Surrogate GCN model $f'_\theta(x)$
Build adjacency $A_{A,2\text{-hop}}$, initialize $F'_{A,2\text{-hop}}$.
for $p_i \in P_{A,2\text{-hop}}$ **do**
  if $p_i \in P_A$ **then**
    Add known x_{p_i} and label y'_{p_i}.
  else
    Collect 1-hop $F^{(1)}$, 2-hop $F^{(2)}$ neighbor features.
    Compute $f^{(1)} = \mathcal{M}_\phi(F^{(1)})$, $f^{(2)} = \mathcal{M}_\phi(F^{(2)})$.
    Synthesize $x'_{p_i} = f^{(1)} + \alpha f^{(2)}$, add to $F'_{A,2\text{-hop}}$.
  end if
end for
Construct updated graph $(F'_{A,2\text{-hop}}, A'_{A,2\text{-hop}})$.
Train 2-layer GCN surrogate $f'_\theta(x)$.

Experiment

The experiments used three citation datasets—Cora (2708 nodes, 5429 edges, 7 classes), Citeseer (3327 nodes, 4732 edges, 6 classes), and PubMed (19717 nodes, 44338 edges, 3 classes)—commonly employed for node classification and network analysis. Nodes represent research papers, edges denote citation links, and classes correspond to paper categories. All data were used to train the victim model, while subsets of 140, 280, 420, 560, and 700 nodes served as labeled attack nodes, with the remainder as unlabeled test nodes. Performance was evaluated using *accuracy* ($f'_\theta(p_i) = y_{p_i}$), defined as correctly classified nodes over total test nodes, and *fidelity* ($f'_\theta(p_i) = f_\theta(p_i)$), defined as the fraction of identical predictions between surrogate and victim models. Accuracy indicates surrogate usefulness for direct inference, while fidelity reflects alignment for adversarial analysis. KL divergence between victim and surrogate logits was also reported to measure probabilistic similarity. The victim was a 2-layer GCN with 32 hidden neurons, ReLU activation, softmax output, Adam optimizer (learning rate 0.01, weight decay 0.0005), trained for 300 epochs. The feature synthesizer $\mathcal{M}_\phi$ was implemented as a two-layer neural network, pretrained on labeled attack nodes and neighbors, and later applied in inference mode during surrogate graph construction (Eq. 1).

	Accuracy			Fidelity		
	Cora	Citeseer	Pubmed	Cora	Citeseer	Pubmed
VN	0.809	0.685	0.790	–	–	–
140	0.772	0.677	0.765	0.856	0.819	0.891
280	0.760	0.678	0.776	0.841	0.803	0.887
420	0.768	0.679	0.777	0.846	0.790	0.890
560	0.775	0.685	0.758	0.838	0.809	0.888
700	0.766	0.673	0.778	0.837	0.801	0.891

Table 2: Proposed attack simulation on all three datasets. $\alpha = 0.5$. Here, VN: victim node.

Evaluation

Fidelity & Accuracy As Table 2 represents the accuracy and fidelity value of the proposed attack simulation across the Cora, Citeseer, and Pubmed datasets. Accuracy measures the performance of the extracted model in replicating the classification ability of the victim model, while fidelity quantifies how closely the extracted predictions of the GNN model are aligned with the victim model. The results show that the proposed attack strategy effectively approximates the target model's decision boundary while maintaining a high level of fidelity. The target model representing the original GCN model trained on the entire dataset achieves an accuracy of 0.809, 0.685, and 0.790 for Cora, Citeseer, and Pubmed, respectively. Yet while varying the number of nodes used in attack scenario, a slight degradation in accuracy is observed, which could be due to the limited knowledge available to the adversary. However, even with only 140 nodes, our proposed attack strategy achieves accuracy values close to the original model, reaching 0.772, 0.677, and 0.765 for Cora, Citeseer, and Pubmed, respectively. As the number of nodes increases, the extracted model maintains relatively stable accuracy, indicating that the attack can successfully infer the behavior of the target model with a small subset of the data. The reliability values further highlight the effectiveness of the proposed attack strategy. Even with just 140 nodes, the extracted model achieves fidelity scores of 0.856, 0.819, and 0.891 across the three datasets, demonstrating a strong agreement with victim model. The fidelity values remain consistently high across different node settings, reinforcing the robustness of the attack. In particular, the highest fidelity is observed in the Pubmed dataset, suggesting that the attack performs exceptionally well on datasets with specific structural properties. Interestingly, fidelity does not always increase with more nodes, suggesting that beyond a certain threshold value, additional nodes do not necessarily improve the attack's ability to replicate the target model. This insight highlights the efficiency of the proposed attack, as it can achieve high fidelity even with a relatively small subset of the graph. To evaluate the effect of the adjustment factor (α) parameter on fidelity, we performed experiments in five attack node settings (140, 280, 420, 560, 700) for three benchmark datasets: Cora, Citeseer, and Pubmed as shown in figure 3. For Cora, fidelity consis-

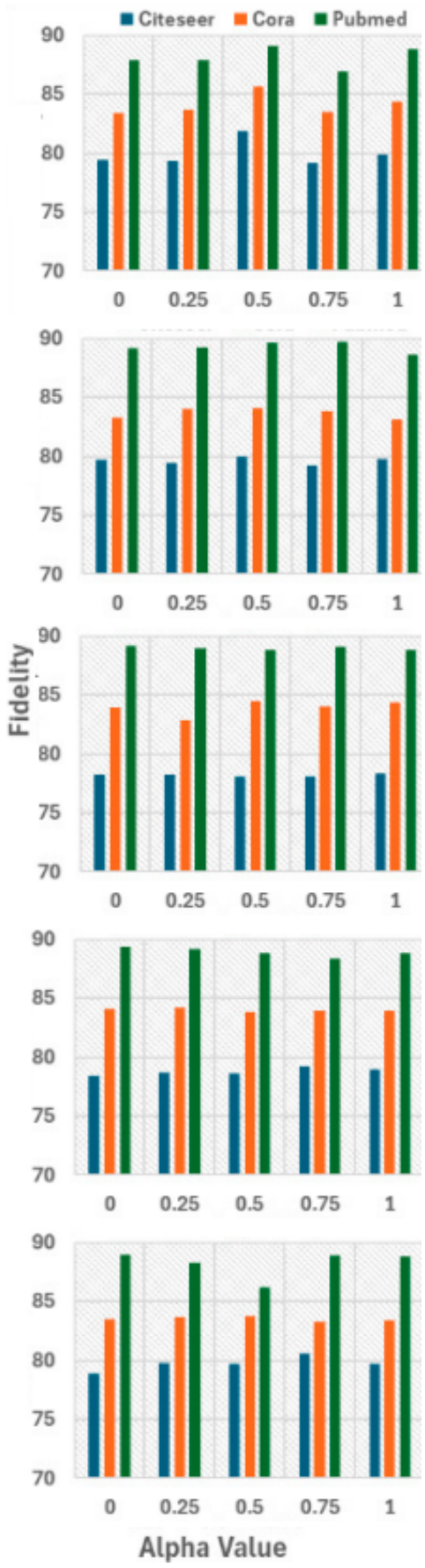

Figure 3: An illustration of fidelity value depending upon the adjustment factor (α), respectively.

tently peaked around α = 0.5 across all attack sizes, indicating that moderate adjustment provides the most effective balance for feature synthesis. Variations in α had a slightly reduced impact at the higher attack nodes, suggesting the robustness of the response of the model on scale. In Citeseer, the fidelity values were generally lower and exhibited minimal alpha sensitivity, with minor improvements observed around α = 0.5–0.75. Notably, at larger attack node counts, alpha had negligible effect, implying data set-specific resistance to perturbation benefits. For Pubmed, fidelity remained consistently high across all values of α and attack nodes. Slight peaks were often observed at α = 0.5 or α = 0.75, but the variation in α remained minimal. This highlights Pubmed's stability and resilience to both attack scale and alpha adjustment.

In general, α = 0.5 was empirically found to be the most effective in data sets and attack scales, particularly Cora and Pubmed. The results also demonstrate that while fidelity is moderately influenced by α at lower attack node counts, its effect diminishes as the number of attack nodes increases.

Soft-label Metric (KL) Although the primary attack operates in a hard-label setting, for a more nuanced understanding of model similarity, we consider an extended soft-label variant where logits are assumed to be accessible (e.g., via logging or leakage). We incorporate Kullback-Leibler (KL) divergence as a secondary fidelity metric to evaluate the proximity of the predictive distributions of the surrogate model to those of the victim model. KL divergence provides a more nuanced perspective on how well a surrogate model captures the probabilistic behavior of the target model, especially in scenarios where prediction distributions may be similar even if top-1 labels differ. The attack itself uses hard labels only. For evaluation, we also report KL divergence in a soft-label variant where logits of the victim and surrogate are observable (e.g., via logging/leakage) purely for analysis. For test nodes S, with probabilities p_v, p_s from softmax over logits (temperature T=1), we compute $KLp_v||p_s$ through the following equation 3.

$$KL(p_v||p_s) = \sum_{i \in S} \frac{1}{|S|} \sum_{c=1}^{C} p_v^{(i)}(c) \cdot \log \frac{p_v^{(i)}(c)}{\max\left(p_s^{(i)}(c), 10^{-12}\right)} \tag{3}$$

We report the victim to surrogate direction to assess how well the surrogate matches the victim's predictive distribution; KL is not used for training. We additionally analyze KL between victim and surrogate predictive distributions under varying α values ranging from 0 to 1 and attack-node counts to probe how feature-synthesis settings affect probabilistic alignment (Figures 4; Table 2).

The KL-divergence in Cora remained relatively stable across α values, with the lowest divergence consistently observed around $\alpha = 0.5$. For instance, at 420 attack nodes, $\alpha = 0.5$ achieved the minimum divergence (0.2666). At higher attack scales (e.g., 700 nodes), divergence decreased further (e.g., 0.2544 at $\alpha = 0.5$), indicating that, with appropriate tuning, moderate smoothing enhances robustness even under larger perturbations. Citeseer showed overall higher

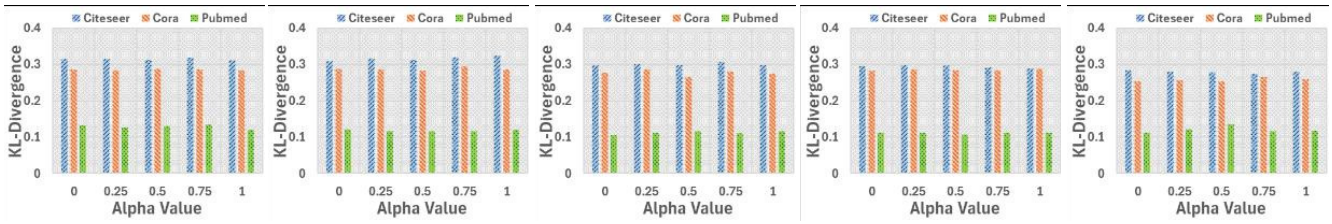

Figure 4: An illustration of non-linear behavior of KL-divergence over node count and adjustment factor (α), respectively.

divergence than Cora and PubMed, ranging roughly from 0.275 to 0.323. The lowest values appeared at $\alpha = 0.5$ or 0.75, particularly at higher numbers of attack nodes; for example, at 700 nodes, $\alpha = 0.75$ yielded the minimum divergence (0.2756). These results suggest that Citeseer's synthesized features become more aligned with the originals under moderate-to-high smoothing when attacks are denser. PubMed exhibited the lowest KL-divergence across all datasets, highlighting strong structural robustness under feature synthesis. At every attack level, divergence remained minimal, with $\alpha = 0.5$ often optimal—for example, at 560 nodes, $\alpha = 0.5$ produced a divergence of just 0.1082. Notably, divergence stayed below 0.14 even at the highest attack setting (700 nodes), underscoring the resilience of PubMed's graph structure. Across all datasets, $\alpha = 0.5$ emerged as the most effective smoothing parameter for minimizing KL-divergence, especially in Cora and PubMed. While Citeseer was more sensitive to α, it still benefited from moderate-to-high values. Overall, these findings reinforce that intermediate smoothing ($\alpha \approx 0.5$) balances feature adaptation and preservation, reducing distributional drift without excessive over-smoothing across varying attack intensities.

Confusion Matrix Figure 5 shows that the Cora dataset has the largest performance gap between the victim (target) and surrogate models. The victim model achieves 81% accuracy with consistently high F1-scores across most classes, particularly in dominant ones such as Class 2 and Class 3. The surrogate model, however, drops to 77% accuracy, with a substantial decline in recall for Class 3 (from 0.77 to 0.63) despite improved precision. This suggests the surrogate model struggles with highly represented or structurally complex classes. While macro and weighted averages remain acceptable, the synthetic feature generation in the surrogate graph appears insufficient to preserve the nuanced topological cues needed for accurate classification in Cora's more intricate structure. For the Citeseer dataset, Figure 5 indicates that the performance gap between the victim and surrogate models is marginal (69% vs. 68%). Both models show relatively consistent F1-scores, with notable stability in Classes 2 through 5. Interestingly, the victim model improves recall for the previously weak Class 0 (from 0.45 to 0.55) at the expense of precision, indicating a more inclusive but less precise classification. This trade-off suggests that Citeseer's flatter class distribution and less complex neighborhood structures are more resilient to feature-synthesis errors, allowing the surrogate to approximate decision boundaries more faithfully than in Cora. In the PubMed dataset, Figure 5 shows strong performance for both models: the victim model reaches 79% accuracy and the surrogate is close behind at 77%. The simple three-class setup and relatively homogeneous graph likely contribute to this high fidelity. The surrogate maintains high precision and recall for the majority class (Class 1), though a slight drop in recall for Class 2 reflects mild degradation. Overall, the smaller class set and clearer structural separation in PubMed enable the surrogate to perform comparably to the victim, indicating that datasets with fewer classes and more regular structures are more amenable to accurate model extraction via surrogate graph construction.

Dataset	Node Count (%)	Fidelity (%)	KL-divergence
Cora	0.5%	49.90	1.647
	1%	70.20	0.674
	5%	**85.29**	0.557
	10%	84.12	0.535
	20%	84.10	0.6119
Citeseer	0.5%	43.40	1.135
	1%	71.61	0.675
	5%	**81.88**	0.5032
	10%	80.10	1.086
	20%	80.76	1.402
PubMed	0.5%	80.40	0.310
	1%	**88.82**	0.377
	5%	88.70	0.416
	10%	87.85	0.328
	20%	88.02	0.297

Table 3: Query (Attack Node) Efficiency: Fidelity and Node Counts

Attack Node (%) Count Each query corresponds to requesting a top-1 label for an attack node. Since neighbors are synthesized, the query count equals the attack nodes. Table 3 shows that even with 10% queries, fidelity exceeds 80% across datasets, and PubMed surpasses 88%. This underscores efficiency in low-query regimes, making the attack practical in ML-as-a-service settings. Monitoring query counts thus emerges as a key defense direction.

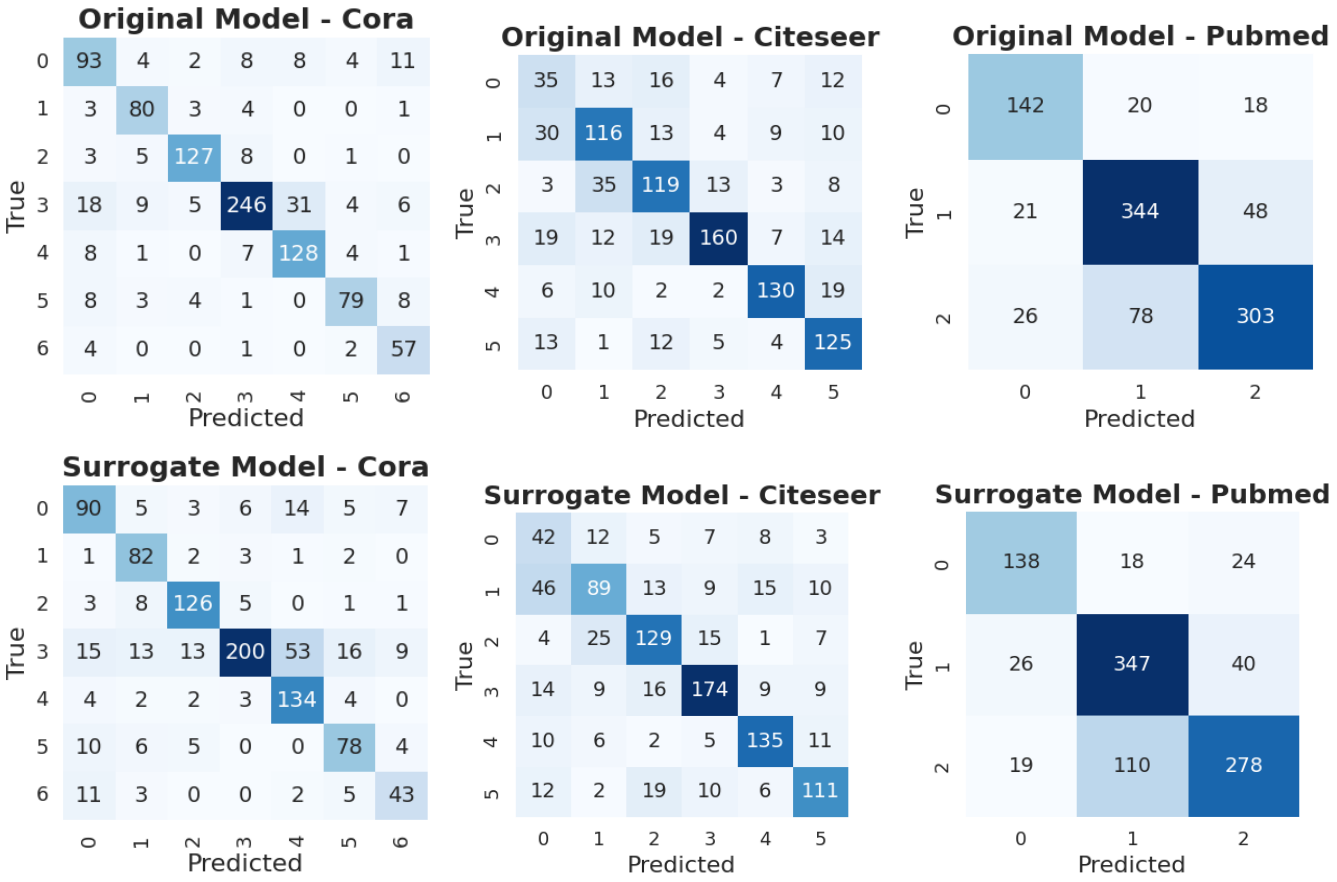

Figure 5: Confusion Matrix for Target Model and Surrogate Model predictions, for each dataset respectively.

Conclusion & Future Work

The respective paper proposes a GNN-based model extraction attack, which includes generating queries as input to the nodes of the target graph, subsequently utilizing known structure knowledge and the responses of queries to clone the functionality of the target model. The proposed attack introduces a model for the synthesis of learnable characteristics that improves the generalization capability of the generation of synthetic node attributes. This allows the attacker to more accurately mimic the victim's GCN behavior using limited feature visibility. Furthermore, evaluation using the KL-divergence provides deeper insight into model alignment beyond label accuracy. In future work, we plan to extend the attack by learning synthetic structure generation using graph completion or generative models. Another direction for future work involves incorporating query-efficiency metrics into the evaluation, such as logging total API calls and computing queries-per-parameter, considering relaxed settings that allow partial introspection or utilize model size estimation techniques to assess cost-effectiveness and query optimization strategies of CPS attack. We could also aim to investigate the impact of surrogate models in downstream adversarial settings, such as crafting transferable attacks (e.g., PGD or NETTACK) using a surrogate model and evaluating their effectiveness on the victim GCN model. This direction can help assess not just model imitation but also the potential security risks stemming from surrogate-driven adversarial attacks.

References

Acharya, D. B.; and Zhang, H. 2024. Feature Selection and Extraction for Graph Neural Networks. ArXiv:1910.10682 [cs].

Ahmed, R.; Bibi, M.; and Syed, S. 2023. Improving Heart Disease Prediction Accuracy Using a Hybrid Machine Learning Approach: A Comparative study of SVM and KNN Algorithms. *International Journal of Computations, Information and Manufacturing (IJCIM)*, 3(1): 49–54.

Chandrasekaran, V.; Chaudhuri, K.; Giacomelli, I.; Jha, S.; and Yan, S. 2018. Exploring Connections Between Active Learning and Model Extraction. Version Number: 6.

Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2018. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. *arXiv preprint arXiv:1810.05997*.

Gong, N. Z.; and Liu, B. 2018. Attribute Inference Attacks in Online Social Networks. *ACM Trans. Priv. Secur.*, 21(1).

He, X.; Jia, J.; Backes, M.; Gong, N. Z.; and Zhang, Y. ???? Stealing Links from Graph Neural Networks.

He, X.; Jia, J.; Backes, M.; Gong, N. Z.; and Zhang, Y. 2020. Stealing Links from Graph Neural Networks. *arXiv preprint arXiv:2005.02131*.

Hong, Z.; Wang, Z.; Shen, L.; Yao, Y.; Huang, Z.; Chen, S.; Yang, C.; Gong, M.; and Liu, T. 2023. Improving Non-Transferable Representation Learning by Harnessing Content and Style. *arXiv preprint arXiv:2303.06403*.

Jagielski, M.; Berthelot, D.; Kurakin, A.; and Papernot, N. ???? High Accuracy and High Fidelity Extraction of Neural Networks.

Juuti, M.; Szyller, S.; Marchal, S.; and Asokan, N. 2018. PRADA: Protecting against DNN Model Stealing Attacks. Version Number: 5.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*.

Li, J.; Zhang, H.; Han, Z.; Rong, Y.; Cheng, H.; and Huang, J. 2020a. Adversarial Attack on Community Detection by Hiding Individuals. Publisher: arXiv Version Number: 1.

Li, S.; Ma, S.; Xue, M.; and Zhao, B. Z. H. 2020b. Deep Learning Backdoors. Version Number: 2.

Milli, S.; Schmidt, L.; Dragan, A. D.; and Hardt, M. 2019. Model Reconstruction from Model Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 1–9. Atlanta GA USA: ACM. ISBN 978-1-4503-6125-5.

Orekondy, T.; Schiele, B.; and Fritz, M. 2019. Knockoff Nets: Stealing Functionality of Black-Box Models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4949–4958. Long Beach, CA, USA: IEEE. ISBN 978-1-72813-293-8.

Pal, S.; Gupta, Y.; Shukla, A.; Kanade, A.; Shevade, S.; and Ganapathy, V. 2019. A framework for the extraction of Deep Neural Networks by leveraging public data. Version Number: 1.

Syed, S.; Ahmed, R.; Iqbal, A.; Ahmad, N.; and Alshara, M. A. 2024a. MediScan: A Framework of U-Health and Prognostic AI Assessment on Medical Imaging. *Journal of Imaging*, 10(12): 322.

Syed, S.; Khan, K.; Khan, M.; Khan, R. U.; and Aloraini, A. 2024b. Recognition of inscribed cursive Pashtu numeral through optimized deep learning. *PeerJ Computer Science*, 10: e2124.

Syed, S.; Syed, Z.; Mahmood, P.; Haider, S.; Khan, F.; Syed, M. T.; and Syed, S. 2023. Application of coupling machine learning techniques and linear Bias scaling for optimizing 10-daily flow simulations, Swat River Basin. *Water Practice & Technology*, 18(6): 1343–1356.

Syed, S.; Talha, S. M.; Iqbal, A.; Ahmad, N.; and Alshara, M. A. 2024c. Seeing Beyond Noise: Improving Cryptocurrency Forecasting with Linear Bias Correction. *AI*, 5(4): 2829–2851.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing Machine Learning Models via Prediction APIs. 601–618. ISBN 978-1-931971-32-4.

Wang, B.; and Gong, N. Z. 2019. Attacking Graph-based Classification via Manipulating the Graph Structure. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2023–2040. London United Kingdom: ACM. ISBN 978-1-4503-6747-9.

Wang, B.; Zhou, T.; Lin, M.; Zhou, P.; Li, A.; Pang, M.; Fu, C.; Li, H.; and Chen, Y. 2020. Evasion Attacks to Graph Neural Networks via Influence Function.

Wu, B.; Yang, X.; Pan, S.; and Yuan, X. 2021a. Adapting Membership Inference Attacks to GNN for Graph Classification: Approaches and Implications. In *2021 IEEE International Conference on Data Mining (ICDM)*, 1421–1426. Auckland, New Zealand: IEEE.

Wu, B.; Yang, X.; Pan, S.; and Yuan, X. 2021b. Model Extraction Attacks on Graph Neural Networks: Taxonomy and Realization. ArXiv:2010.12751 [cs].

Xu, J.; Xue, M. J.; and Picek, S. 2021. Explainability-based Backdoor Attacks Against Graph Neural Networks. In *Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning*, 31–36. Abu Dhabi United Arab Emirates: ACM. ISBN 978-1-4503-8561-9.

Zhang, H.; Wu, B.; Yang, X.; Zhou, C.; Wang, S.; Yuan, X.; and Pan, S. 2021a. Projective Ranking: A Transferable Evasion Attack Method on Graph Neural Networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 3617–3621. Virtual Event Queensland Australia: ACM. ISBN 978-1-4503-8446-9.

Zhang, Z.; Jia, J.; Wang, B.; and Gong, N. Z. 2021b. Backdoor Attacks to Graph Neural Networks. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, 15–26. Virtual Event Spain: ACM. ISBN 978-1-4503-8365-3.

Zhu, S.; Pan, S.; Zhou, C.; Wu, J.; Cao, Y.; and Wang, B. 2020. Graph Geometry Interaction Learning. In *Advances in Neural Information Processing Systems*, volume 33, 7548–7558. Curran Associates, Inc.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial Attacks on Neural Networks for Graph Data. Publisher: arXiv Version Number: 4.