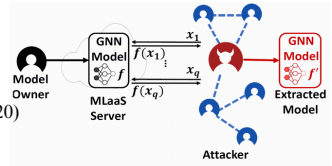
CEGA: A Cost-Effective Approach for Graph-Based Model Extraction Attacks

Ken Anderson, Kien Le and Dr. Yushun Dong
Florida State University Department of Computer Science

Introduction

Graph Neural Networks (GNNs) are a powerful machine learning approach designed to analyze graph-structured data. By leveraging the connectivity of graphs, GNNs demonstrate their effectiveness in prediction tasks for social networks, biology, and finance. Such models, however, are expensive to train, leading companies to offer them as Machine Learning as a Service (MLaaS), allowing users to access GNNs via a pay-per-query system. But, this creates a security risk: adversaries can strategically query a GNN model to recreate its functionality through a Model Extraction Attack (MEA). In this poster, we investigate a MEA scenario where the attacker has limited knowledge of the target GNN model and apply transferable active learning to reduce the number of queries required to build a comprehensive surrogate model. Through experiments on multiple datasets, preliminary results show our approach achieves high fidelity and accuracy while adhering to strict query constraints.



(Wu et al., 2020)

Figure 1: Basic illustration of a MEA. A model owner provides a GNN model and the service of prediction queries. An attacker extracts a surrogate model based on the answers from the server.

Inspiration

The figure below illustrates our inspiration for using GPA in a MEA scenario. Compared to another active learning baseline, AGE (Cai et al., 2017), we can see how GPA selects a more diverse set of nodes. We hope that GPA can enhance the node selection process in a MEA by strategically choosing the most informative nodes to query, thereby reducing the number of labeled nodes needed to train a surrogate model.

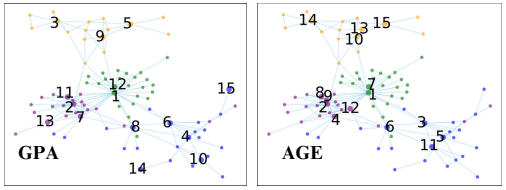


Figure 2: Visualization of the node query process for a Reddit graph using GPA and AGE. The query budget is 15. (Hu et al., 2020)

Experiment Setup

We apply active learning on graphs using the method outlined in the paper, "**G**raph **P**olicy Network for Transferable **A**ctive Learning on Graphs", (GPA). GPA selects the most informative nodes for labeling in a graph by using a reinforcement learning based policy network. By training on multiple fully labeled graphs, it learns a strategy to maximize information gain by iteratively selecting nodes that improve a GNN model's performance. This trained policy can be generalized to various unlabeled graph structures.

Our MEA scenario follows Attack-0 outlined in (Wu et al., 2020) which operates under the constraint that only partial node attributes and partial graph structure are known. The key steps of Attack-0 is as follows: (1) Randomly select attack nodes from the target graph, (2) construct an attack graph graph by synthesizing node attributes using information from 1-hop and 2-hop neighbors, (3) query the victim GNN model to obtain labels for the selected attack nodes, (4) train a surrogate GNN model using the labeled attack graph, and (5) repeat the process iteratively until all attack nodes have been labeled, allowing the surrogate model to approximate the victim model's predictions.

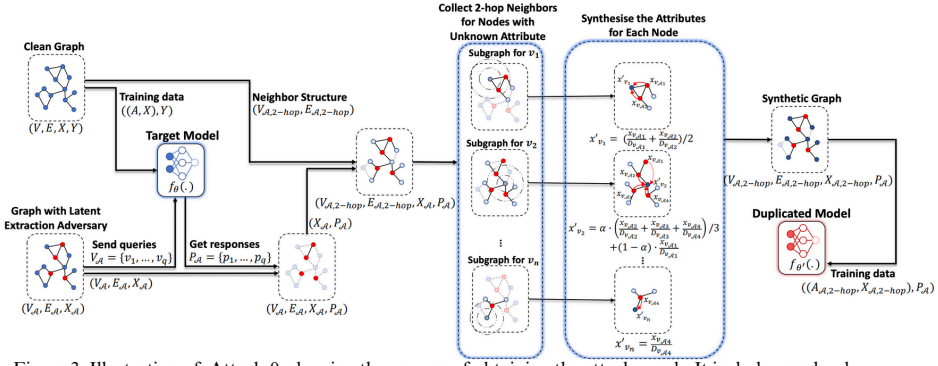


Figure 3: Illustration of Attack-0, showing the process of obtaining the attack graph. It includes randomly selecting attack nodes, creating a subgraph, and synthesizing attribute information. (Wu et al., 2020)

Instead of querying every attack node for a label as done in Attack-0, we integrate a policy created by GPA into Attack-0 to optimize node selection and minimize the number of queries needed to construct a surrogate GNN model. Our experimental setup is as follows:

Step 1: Create a policy for choosing the most informative nodes to query using multiple fully-labeled graphs (Pubmed and Citeseer).
Step 2: Train a victim GNN model used to predict the labels of a Cora dataset.
Step 3: Randomly select a limited number of attack nodes from the target graph (Cora).
Step 4: Connect all of the attack nodes and synthesize attribute information by combining information from 1-hop and 2-hop neighbor node information.
Step 5: Using the learned policy created from Step 1, choose the most informative node to query given the current state of the attack graph.
Step 6: Query the victim GNN to obtain a prediction. The label predicted by the target model will be used to label the node in the attack graph.
Step 7: Update the attack graph state after labeling the node and train the surrogate model once.
Step 8: Repeat steps 5-7 until the query budget is reached.
Step 9: Train the surrogate model 35 more times (until convergence).
Step 10: Compare the results of the surrogate model to the victim GNN model. Compare the performance difference between querying every attack node vs querying under a constrained budget.

Results

Metrics	Accuracy	Fidelity	F1
GPA	72.47	75.47	70.18
Random	74.03	78.23	72.31
AGE	72.38	76.38	70.23
CEGA	75.73	80.38	74.14

Table 1: Test accuracy, fidelity, and F1 scores on the Cora dataset using a budget of 20C queried nodes.

In Table 1, we observed that GPA performed on par or slightly worse than the other benchmarks on the Cora dataset. In this context, *accuracy* measures how close predicted labels are to the ground truth labels, *fidelity* measures the similarity between the surrogate model's predictions and those of the victim model, and *f1* evaluates the balance between precision and recall.

Compared to AGE, GPA only performed better in accuracy. Interestingly, both active learning methods, AGE and GPA, scored lower on all three metrics compared to the random selection baseline. To further evaluate GPA's effectiveness in selecting nodes in a MEA scenario, we will continue testing on different datasets to assess whether these trends persist across varied graph structures and node distributions.

A QR code for updated results:

Acknowledgement

We would like to express our sincere gratitude to Dr. Yushun Dong, Zebin Wang, and Menghan Lin for their invaluable mentorship and guidance throughout this research. We also extend our appreciation to the Florida State University's Department of Computer Science and the Center for Research Engagement for providing the resources and support necessary to conduct this project.

Resources