Graph Backdoor

Zhaohan Xi\textsuperscript{1}, Ren Pang\textsuperscript{1}, Shouling Ji\textsuperscript{2}, Ting Wang\textsuperscript{1}

\textsuperscript{1}Pennsylvania State University, College of Information Science and Technology
\textsuperscript{2}Zhejiang University, College of Computer Science and Technology
Motivation

- **Backdoor attacks against DNNs**
  - A trojan model responds to trigger-embedded inputs in a specific manner
  - While the trojan model functioning normally for untouched inputs

- **Graph data and GNNs**
  - Graph data format is widely use as a flexible representation
  - GNNs are learning-based models to capture graph/node properties
  - The vulnerabilities in graphs and GNNs are largely unexplored

- **Graph-domain challenges**
  - **Trigger definition**: has both topological structure and descriptive features
  - **Input-tailored**: a trigger is tailored to the characteristics of an individual graph
  - **Adaptive location**: a trigger should be embedded into a suitable locality
GTA: Graph Trojaning Attack

- **Upstream: adaptive learning**
  - The adversary forges a trojan GNN $f_\theta$ (pre-trained model) via perturbing its parameters
  - To realize attack, the adversary leverages bi-level optimization between $f_\theta$ and trigger $g_t$

- **Downstream: model-agonistic**
  - The adversary has no access to downstream model $h$, but $z_G$ can lead to a falsified result
GTA: Trigger Generation

**Graph encoding**
- Use attention nets to encode $G$ and get $Z$
- The encodings are assured to capture both topological information and original features

**Trigger generation**
- Node connectivity: $\tilde{A}_{ij} = \mathbb{I}_{\text{sim}(\phi_\omega(z_i), \phi_\omega(z_j))} \geq 0.5$
- Backdoor features: $\tilde{X}_i = \sigma(Wz_i + b)$, $W, b \in \phi_\omega$
- Combine $\tilde{A}$ and $\tilde{X}$ as $g_t$, where $i, j \in g_t$
GTA: Backdoor Poisoning

Trigger Injection
- Rely on mixing function $m(G; g_t)$
  - Find to-be-replaced subgraph $g \in G$
  - Substitute $g$ with $g_t$

Backdoor Poisoning
- Inject trigger to not-target-label graphs $\mathcal{D}_{[\text{notarget}]}$
- Train GNNs $\theta$ with poisoned set $\mathcal{D}$
GTA: Bi-level Optimization

- **Upper level – optimize trigger**
  - \( g_t^* = \arg \min_{g_t} l_{atk}(\theta^*(g_t), g_t) \)
  - \( l_{atk} \): difference between \( g_t \)-embedded graphs and \( G \in \mathcal{D}_{y_{tar}} \) through GNNs

- **Lower level – optimize GNNs**
  - \( \theta^*(g_t) = \arg \min_{\theta} l_{ret}(\theta, g_t) \)
  - \( l_{ret} \): loss of GNNs
**Evaluation Settings**

- **Multi-domain dataset**
  - Security-sensitive domains
  - Biology and chemistry
  - Social and transaction networks

- **Manifold learning settings**
  - Inductive (graph-level) & transductive (node-level) classification
  - Self-transfer & mutual-transfer learning
  - Graph-space (default) & input-space attacks

| Dataset       | Domain            | Setting                        | # Samples         |
|---------------|-------------------|-------------------------------|-------------------|
| Fingerprint   | Cybersecurity     | Inductive, self-transfer      | 1.6k graphs       |
| WinMal        | Cybersecurity     | Inductive, self-transfer      | 1.3k graphs       |
| AIDS          | Biochemistry      | Inductive, mutual-transfer    | 2.0k graphs       |
| Toxicant      | Biochemistry      | Inductive, mutual-transfer    | 10.3k graphs      |
| AndroZoo      | Cybersecurity     | Inductive, input-space        | 0.2k graphs       |
| Bitcoin       | Transaction net   | Transductive                  | 5.6k nodes        |
| Facebook      | Social net        | Transductive                  | 12.5k nodes       |
Evaluation Settings (cont.)

- **Representative GNNs**
  - GCN (Kipf & Welling, 2017)
  - GAT (Velickovic et al. 2018)
  - GraphSAGE (Hamilton et al. 2017)

- **Self-variant baselines**
  - $BL^I$: a universal trigger with fully connected topo. + adaptive features
  - $BL^{II}$: a universal trigger with adaptive topo. + adaptive features

- **Comprehensive metrics**
  - Effectiveness: attack success rate (ASR), etc.
  - Evasiveness: clean accuracy drop (CAD), etc.

| Dataset            | GNN   | Benign Acc. |
|--------------------|-------|-------------|
| Fingerprint $\mathcal{U}$ | GAT   | 82.9%       |
| WinMal $\mathcal{U}$    | GraphSAGE | 86.5%    |
| Toxicant $\rightarrow$ AIDS | GCN   | 93.9%       |
| AIDS $\rightarrow$ Toxicant | GCN   | 95.4%       |
| ChEMBL $\rightarrow$ AIDS | GCN   | 90.4%       |
| ChEMBL $\rightarrow$ Toxicant | GCN   | 94.1%       |
| AndroZoo (A.)        | GCN   | 95.3%       |
| AndroZoo (A.+F.)     | GCN   | 98.1%       |
| Bitcoin              | GAT   | 96.3%       |
| Facebook             | GraphSAGE | 83.8%    |

- Abbreviation: A. – only use topology; A.+F. – use both topology and raw features
# Evaluations

## Inductive settings

| Settings          | $\mathbf{BL}^I$     | $\mathbf{BL}^{II}$       | GTA       |
|-------------------|---------------------|---------------------------|-----------|
|                   | ASR, CAD            | ASR, CAD                  | ASR, CAD  |
| Fingerprint $\mathcal{F}$ | 84.4%, 1.9%         | 87.2%, 1.6%               | 100%, 0.9%|
| WinMal $\mathcal{F}$ | 87.2%, 1.8%         | 94.4%, 1.2%               | 100%, 0.0%|
| Toxicant $\rightarrow$ AIDS | 89.4%, 1.7%         | 95.5%, 1.3%               | 98.0%, 1.4%|
| AIDS $\rightarrow$ Toxicant | 80.2%, 0.6%         | 85.5%, 0.0%               | 99.8%, 0.4%|

## Use the off-the-shelf GNNs

| Settings          | $\mathbf{BL}^I$     | $\mathbf{BL}^{II}$       | GTA       |
|-------------------|---------------------|---------------------------|-----------|
|                   | ASR, CAD            | ASR, CAD                  | ASR, CAD  |
| ChEMBL $\rightarrow$ AIDS | 92.0%, 1.1%         | 97.5%, 1.0%               | 99.0%, 1.2%|
| ChEMBL $\rightarrow$ Toxicant | 83.5%, 0.6%         | 86.0%, 0.0%               | 96.4%, 0.1%|
### Evaluations (cont.)

- **Transductive settings (node-level classification)**

| Settings | BL I | BL II | GTA |
|----------|------|-------|-----|
| ASR, CAD | ASR, CAD | ASR, CAD |
| Bitcoin | 52.1%, 0.9% | 68.6%, 1.2% | 89.7%, 0.9% |
| Facebook | 42.6%, 4.0% | 59.6%, 2.9% | 69.1%, 2.4% |

- **Downstream model agnostic (different classifiers)**

| Classifiers       | BL I      | BL II     | GTA       |
|-------------------|-----------|-----------|-----------|
| ASR, CAD          | ASR, CAD  | ASR, CAD  | ASR, CAD  |
| Naïve Bayes       | 87.7%, 1.5% | 92.4%, 0.9% | 99.5%, 0.7% |
| Random Forest     | 85.8%, 0.9% | 88.0%, 0.9% | 90.1%, 0.6% |
| Gradient Boosting | 82.5%, 0.6% | 89.3%, 0.6% | 94.0%, 0.6% |
Input-space Case Study

- **Input-space constraints**
  - Transferable perturbations (triggers) from graph space
  - Not affect original functionalities of raw data samples
  - If possible, not incur observable semantic variations

- **GTA against Android Malware Detector (GNN-based)**

| Settings            | Input-space GTA | Graph-space GTA |
|---------------------|-----------------|-----------------|
|                     | ASR  | CAD  | ASR  | CAD  |
| Topology Only       | 94.3% | 0.9% | 97.2% | 0.0% |
| Topology + Feature  | 96.2% | 1.9% | 100%  | 0.9% |

Android Call Graph

(a) Original graph locality

(b) Trigger-embedded graph
Potential Countermeasures

- **Data inspection: Randomized Smoothing (Zhang et al. 2020)**
  - Subsample a (possibly trigger-embedded) graph $G$ and generate $G_1, G_2, \ldots, G_n$
  - Take a majority voting among $G_1, G_2, \ldots, G_n$ as $G$’s final classification results
  - Adjust subsample ratio $\beta$ on both of node set and feature dimensions

- **Model inspection: Neural Cleanse (Wang et al. 2019)**
  - For each label, learn a reversed trigger from a backdoored GNN
  - Get the perturbation scale ($L_1$-norm) between the original graphs and the trigger-embedded
  - Use statistical approaches to measure which label has minimum perturbation scale
  - Consider different adaptiveness of reversed trigger (same as $BL^I$ and $BL^{II}$)
Summarizations

- **Graph-oriented**
  - GTA defines a trigger as a subgraph, including topo. structure and descriptive features

- **Input-tailored**
  - GTA generates triggers tailored to the characteristics of individual graphs

- **Downstream-model-agnostic**
  - GTA has no assumption of downstream model (used classifiers), leads to resistive trojaining attack

- **Attack-extensible**
  - GTA represents an attack framework on both inductive and transductive learning settings
Thank You!

For questions, feel free to contact

zxx5113@psu.edu