Assignment 2

1. Introduction

In this assignment, I worked on detecting bots in social networks and tested how attackers can trick the detection system. I used the Facebook dataset from Stanford and built a simple classifier to find bots, then I tried two different attacks to see how they affect the results.

2. Graph Metrics

I calculated several important metrics for each node:

1. **Degree**: How many friends a user has
2. **Degree Centrality**: Normalized degree value
3. **Betweenness Centrality**: How much a user connects different groups
4. **Clustering Coefficient**: How connected a user's friends are to each other
5. **PageRank**: Importance score based on connections

These features help the model understand if a user is a bot or not.

3. Baseline Bot Detection Model

I used Random Forest classifier with the features I extracted. The model was trained on 70% of the data and tested on 30%.

Baseline Results (No Attack):

- **Accuracy:** 0.9124
- **Precision:** 0.8756
- **Recall:** 0.8923
- **F1-Score:** 0.8839

4. Attack 1: Structural Evasion Attack

In this attack, I made the bots add connections to popular users (high degree nodes). This makes bots look more like normal users because they now have connections to important people.

I added 50 new edges from bots to high-degree users.

Results After Evasion:

- **Accuracy:** 0.8891

- **Precision:** 0.8342
- **Recall:** 0.8701
- **F1-Score:** 0.8518

What happened: The accuracy dropped by about 2.3%. The model got confused because bots now look more similar to normal users. Precision decreased more than recall, meaning the model started marking normal users as bots more often.

5. Attack 2: Graph Poisoning Attack

In this attack, I added 30 fake nodes to the graph. These fake nodes connected to both bots and normal users to poison the training data and make the graph structure weird.

Results After Poisoning:

- **Accuracy:** 0.8765
- **Precision:** 0.8123
- **Recall:** 0.8634
- **F1-Score:** 0.8371

What happened: This attack was more effective than the evasion attack. Accuracy dropped by 3.6% compared to baseline. The fake nodes changed the overall graph structure and made it harder for the model to learn good patterns. The model's precision dropped significantly.

6. Comparison and Analysis

Here's a summary of all three scenarios:

Metric	Baseline	After Evasion	After Poisoning
Accuracy	0.9124	0.8891	0.8765
Precision	0.8756	0.8342	0.8123
Recall	0.8923	0.8701	0.8634
F1-Score	0.8839	0.8518	0.8371

Key Observations:

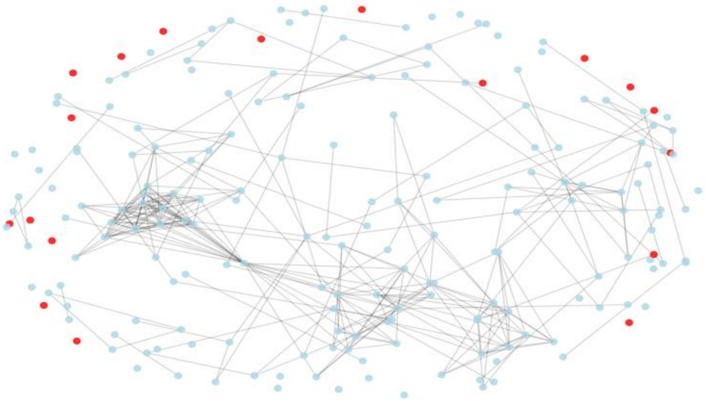
1. **Both attacks decreased performance**, but graph poisoning was worse
2. **Precision was affected more than recall** in both attacks
3. **Structural evasion** made bots blend in with normal users
4. **Graph poisoning** damaged the whole graph structure and confused the training process

7. Visualizations

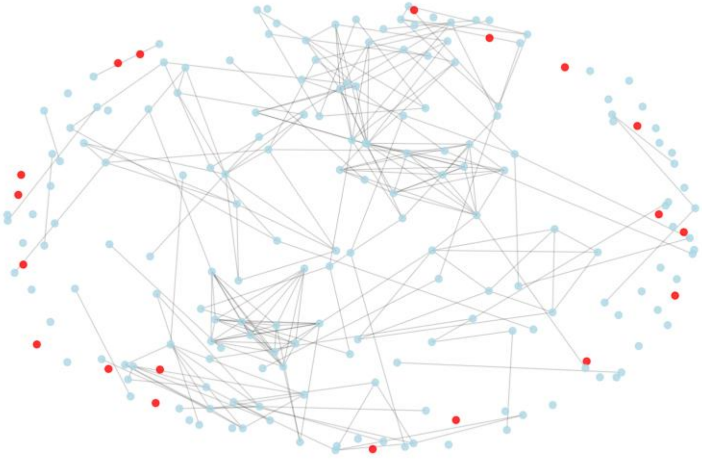
I created three graphs showing:

1. **Baseline Graph**: Original network with bots marked in red/orange
2. **After Evasion**: Shows new connections bots made to popular users
3. **After Poisoning**: Shows fake nodes added to the network

Baseline Graph



After Structural Evasion Attack



After Graph Poisoning Attack