Jana Adel Ramadan 2205105

Social network

Assignment 2

1. Introduction
   Get a real graph from SNAP, Set up a bot detection system, Attack the system and see if it crashes, Compare the before and after results
2. Download data
   Use facebook_combined.txt.gz .

3. Facebook's SNAP data doesn't contain any bots, We'll designate a few nodes to be bots to train the detector and attack it. Because the data doesn't contain bots, we'll create bots, and everything else will be humans.
4. Extracting Features for Each Node

Any detector must have data, we have a graph, so we'll extract the following properties:

- degree: The number of edges per node sometimes the most important feature for bots.
- clustering: The local clustering coefficient reflects how connected the node's neighbors are.
- triangles: The number of triangles in the node (how many triangles pass through it).
- pagerank: A popular measure of centrality; multiple attempts may fail if the network is very large or disconnected, hence the try/except method.
- betweenness: Inter-centrality computationally expensive, so if the network is large (>3000), we use an approximation with k samples (measuring the shortest paths).
- avg_neighbor_deg: The average neighbor degree gives a picture of the node's perimeter.
- core_number: Coreness (k-core) an indicator of the node's connection to the network's core.

These are the features the model will train on.

5. Baseline، Evasion and Poisoning Models
   1. Building a baseline bot detector:

Take the features and labels and train the classifier: Random Forest.

The baseline model is the model before the attack.

record the performance: Accuracy, Precision, Recall, F1-score, support.

2. Structural Evasion Attack

The bot itself "masquerades", modify the connections around the bot node, add 3 human friends to it, expanded its clusterin, makes its degree more human-like, The bot tries to appear human.

After modifying the edges around the bots: Recalculate the features, Retest the model.

3. Graph Poisoning Attack

This is the most dangerous attack: We don't change the bot's appearance itself, We corrupt the entire graph before training.

For example: We add 7% random edges that mimic human behavior to the bots, we change the community structure, Or we add fake nodes.

Then: We retrain the model, We test it on a clean graph.

You'll find that the accuracy is severely compromised because the model was poisoned during training.

```
...
    Baseline classifier report:
                  precision    recall  f1-score   support

               0       1.00      1.00      1.00      1127
               1       1.00      0.98      0.99        85

        accuracy                           1.00      1212
       macro avg       1.00      0.99      0.99      1212
    weighted avg       1.00      1.00      1.00      1212

    Baseline ROC AUC: 0.9999791220836161
```

```
...
    Poisoned-training evaluation report (test on clean):
                  precision    recall  f1-score   support

               0       0.94      1.00      0.97      1127
               1       0.95      0.22      0.36        85

        accuracy                           0.94      1212
       macro avg       0.95      0.61      0.67      1212
    weighted avg       0.95      0.94      0.93      1212

    Poisoned ROC AUC (test on clean): 0.8777597995720027
```

```
...
    Evasion evaluation report:
                  precision    recall  f1-score   support

               0       0.98      1.00      0.99      1127
               1       0.98      0.68      0.81        85

        accuracy                           0.98      1212
       macro avg       0.98      0.84      0.90      1212
    weighted avg       0.98      0.98      0.97      1212

    Evasion ROC AUC: 0.9974894305548305
```

6. Graph Visualization