Graph-based Bot Detection + Attacks on SNAP Facebook Egonets

1. Experiment Setup & Baseline

First we loaded the anonymized SNAP Facebook Egonets dataset (combined graph) — which has **4039 nodes** and **88234 edges** as an undirected graph

We then computed a set of standard graph-structural features for each node:

- Degree
- Clustering coefficient
- Degree centrality
- Betweenness centrality
- Eigenvector centrality

Because the dataset does *not* come with “bot” / “normal user” labels, I simulated a binary classification task by randomly assigning **~5% of nodes (201 / 4039)** as “synthetic bots” (label = 1), and the rest as “normal users” (label = 0).

Using these features I trained a baseline classifier (Random Forest Classifier) to detect “bots” vs “normal.”after the test , I obtained:

- Very high accuracy (~ 94.9%)
- But **very poor performance on the “bot” class**: precision ≈ 0.33, recall ≈ 0.03, F1 ≈ 0.06
- ROC AUC ≈ 0.496

Interpretation:

- The high accuracy is misleading: because the data is highly imbalanced (only ~5% bots), the classifier can achieve ~95% accuracy simply by labeling nearly everyone as “normal.”
- The very low recall / F1 on bots means the model fails to detect most of the synthetic bots under this baseline.

Conclusion: with this naive feature set and synthetic labeling, the baseline detector is effectively useless for identifying bots.

2. Structural Evasion Attack — Results & Meaning

I applied a **structural-evasion attack**: for each synthetic bot node, I randomly removed and rewired up to 10 edges (rewiring their “friendship” connections).

Then re-computed graph features on the resulting graph and ran the original classifier (without retraining) again.

After attack, you observe:

- The classifier predicts **none** of the bots (0 / 201) as bots — recall = 0, precision = 0. So the model fails completely on the positive class
- Accuracy remains ~ 94.8% (because normal users remain the large majority).
- Macro-average F1 drops to ~ 0.486.
- The ROC AUC increases to ~ 0.585 (from 0.496).

Interpretation – What happened & why:

- The structural evasion caused the “bot” nodes to rewire their connections so that their structural-feature vectors likely now resemble “normal” users. As a result, the classifier — which was relying on such features — no longer recognizes them as anomalous, so it classifies all of them as normal.
- The AUC improvement (from ~0.496 to ~0.585) is misleading: while the ranking metric improved, the actual detection of “bots” failed (recall 0). This suggests that under some classification thresholds the model may appear “better,” but in practice it misses all adversarial bots.
- In short: the evasion attack completely defeats the simple graph-feature detector.

structural attacks on graph-based anomaly detection can drastically degrade detection effectiveness by disrupting graph structure.

---

3. Graph Poisoning Attack — Results & Meaning

For the **graph-poisoning attack**: i randomly flipped 2000 edges (some removed, some added) across the entire graph (not just bots). After this disturbance i recomputed features and ran the same classifier. Results:

- Bot detection remains very poor: only ~ 5% of bots are detected (precision 0.125, recall 0.005, F1 ~0.0096).
- Overall accuracy is basically unchanged (~ 94.9%).
- ROC AUC increases to ~ 0.656.

Interpretation:

- Random edge flips across the graph have also degraded the detection ability. The classifier still misses almost all bots, though a tiny fraction are caught.
- The increase in ROC AUC (to ~0.656) is perhaps even more misleading: under random perturbations, the ranking of nodes by “bot-likelihood” may shift, but that doesn’t lead to useful detection performance under typical thresholds.
- Graph poisoning, even random, can significantly confuse graph-feature-based detectors: by globally disturbing the structural patterns (community structure, centralities, clustering), the features no longer reflect the original network, and the detector performance collapses.

---

4. What This Tells Us — Strengths, Weaknesses, and Lessons Learned

Strengths of this experiment

- It shows that even a **simple graph-feature based detector** — which might look promising (due to high accuracy) if you only examine overall accuracy — **is extremely fragile** when facing structural modifications (whether targeted rewiring or global poisoning).
- The experiment mirrors the core insight of research on graph adversarial attacks: graph-based anomaly detection is vulnerable because attackers can manipulate structural relations.

Limitations

- The “bot” labels are **synthetic**: you randomly pick 5% of nodes as bots. That means your baseline and attack evaluation are purely hypothetical — they do not reflect how real bots behave. The results only show *relative degradation under attacks*, not actual detection of real malicious users.
- The classifier is extremely naive. Real-world bot-detection systems might use richer features (node attributes, temporal activity, embeddings), not just simple structural metrics. Those might be more robust.
- Because labels are synthetic and random, results (e.g. ROC AUC) may be unstable across different random seeds or different bot-assignment fractions.
