Bot mitigation
Using graph-based machine learning approach

By: Gagan Ganapathy
(IIT2016038)

Mentor: Dr. Satish Kumar Singh

What is bot mitigation ?

Bot mitigation is far more than just identifying your bot traffic. After all, not all bots are bad.

- **Good bots**: the ones we rely on—such as bots that search for and find things on the internet.
- **Bad bots**: ones that hoard resources, perform account takeovers and credential stuffing, launch DDoS attacks, or steal intellectual property.

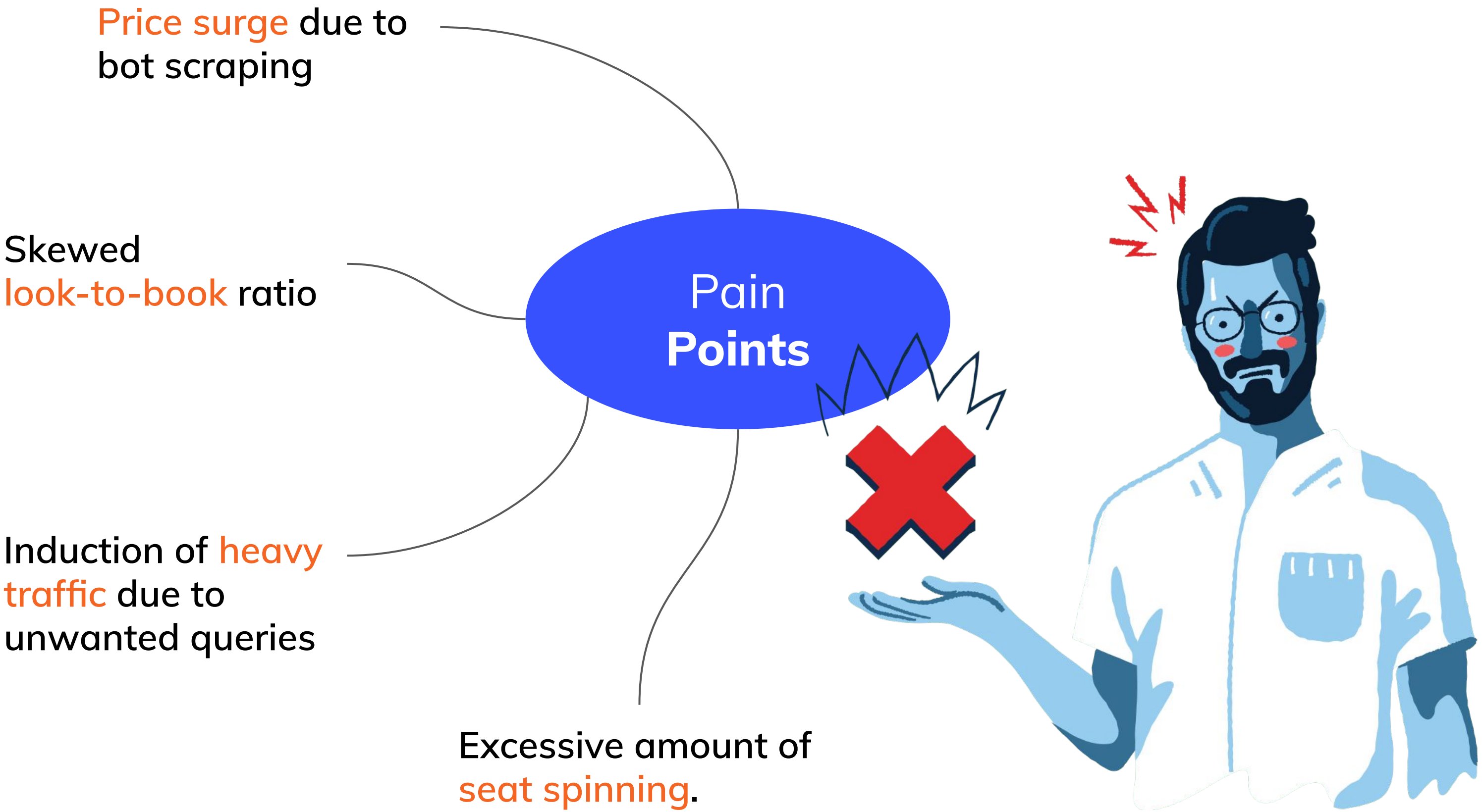
Bot mitigation is about identifying and blocking the unwanted or malicious bot traffic that hits your application or network, so you can reduce your risk.

Why bot mitigation is critical ?

The majority of threats in any environment start with bots or botnets—they help cybercriminals achieve scale. So, It is critical, when looking at your overall security strategy, you consider how you'll mitigate malicious bots.

Industries with the most potential for monetary gain are the hardest hit by bad bots. The top three bot-targeted industries are:

- Gambling sites
- **Airlines and tickets sites**
- Financial institutions



Pain **Points**

Price surge due to bot scraping

Skewed look-to-book ratio

Induction of heavy traffic due to unwanted queries

Excessive amount of seat spinning.

Reference : https://www.infisecure.com/blogs/impact-bad-bots-online-travel-websites

State of the world

84.3%

Bots on OTAs are moderate or advanced which are difficult to detect.

43%

Bad bots traffic on travel websites.

30%

Domains reviewed, Bad bots comprise more than half of all traffic.

Reference : [*https://bit.ly/2qZISBP*](https://bit.ly/2qZISBP)

Previous Solutions to Bot Detection

Achieved using IDS (Intrusion detection systems), broadly classified into **signature** and **anomaly** based.

Signature-based : Uses pre-computed hashes of existing malware binaries. Can be easily subverted by unknown or modified attacks.

Anomaly-based: Overcomes these limitations, establish a baseline of normal behavior for the protected system. Machine learning (ML) is an ideal technique to automatically capture the normal behavior of a system.

Our Solution

- We propose a two-phased, graph-based bot detection system which leverages both unsupervised and supervised ML that accommodates different network topologies and is suitable for large-scale data.

PHASE 1:

Prunes benign hosts. (Unsupervised Learning)

PHASE 2:

Achieves bot detection with high precision. (Supervised Learning)

Why use graph-based features ?

An important step prior to learning, or training a ML model, is feature extraction!

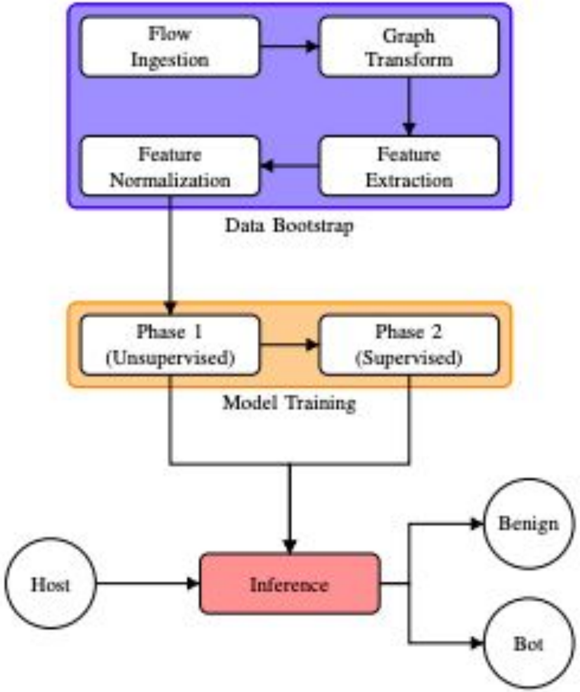
The most commonly employed features in bot detection are flow-based (e.g., **source** and **destination IPs, protocol, number of packets sent and/or received** etc.)

However, these features **do not** completely capture the communication patterns that can expose additional aspects of malicious hosts.

Graph-based features, derived from flow-level information to reflect the true behaviour of hosts!

Moreover, it yields robustness against complex communication patterns and unknown attacks and allows for cross-network ML model training and inference.

Proposed Solution



Feature Extraction

- In-Degree (ID) - f(0)
- Out-Degree (OD) - f(1)
- In-Degree Weight (IDW) - f(2)

- ~~Betweenness Centrality (BC)~~
- Out-Degree Weight (ODW) - f(3)
- Local Clustering Coefficient (LCC) - f(4)

Then the features are normalized by using neighborhood relativity

Dataset

Our evaluation is based on the CTU-13 dataset. CTU-13 comprises of 13 different subset datasets (DS) that include captures from **7 distinct malware**, performing port scanning, DDoS, click fraud, spamming etc.

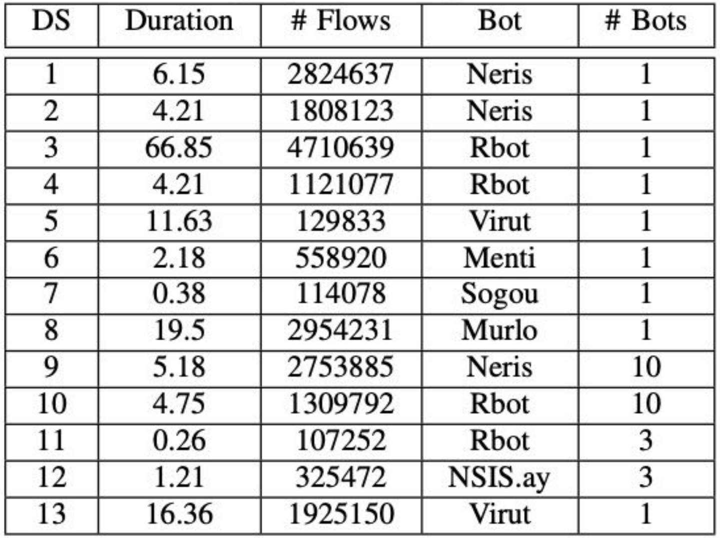
We leverage 7 datasets

(1, 2, 5, 6, 7, 11, and 12) for training.

And,

used the datasets **(9 and 10)** for testing

TABLE I
CTU-13 DATASET



DS	Duration	# Flows	Bot	# Bots
1	6.15	2824637	Neris	1
2	4.21	1808123	Neris	1
3	66.85	4710639	Rbot	1
4	4.21	1121077	Rbot	1
5	11.63	129833	Virut	1
6	2.18	558920	Menti	1
7	0.38	114078	Sogou	1
8	19.5	2954231	Murlo	1
9	5.18	2753885	Neris	10
10	4.75	1309792	Rbot	10
11	0.26	107252	Rbot	3
12	1.21	325472	NSIS.ay	3
13	16.36	1925150	Virut	1

Dataset

Considering all the datasets the ratio to **non-bot entries** and **bot entries** in the dataset is largely skewed and **unbalanced**.

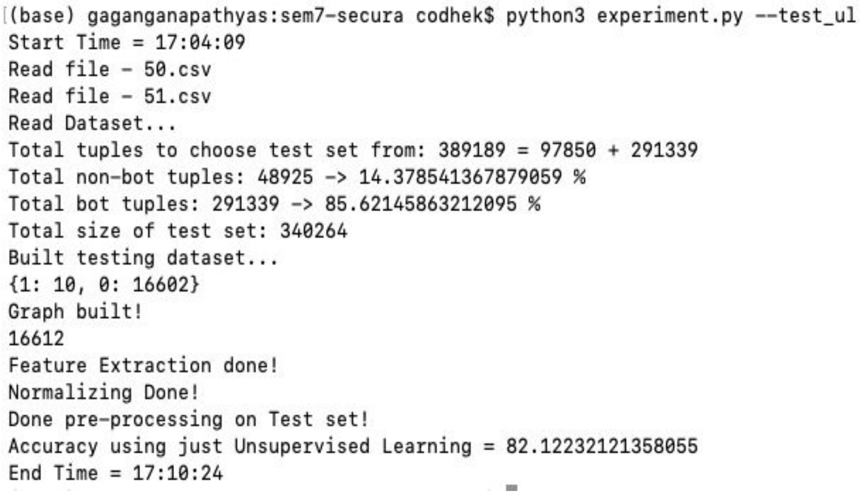
The dataset was modified to balance the two classes :

Initially percentage of bot entries were just **3.98%** of total entries after modification it was brought to **78.68%**

This helps the machine learning algorithms to learn the bot's pattern better and produce better results!

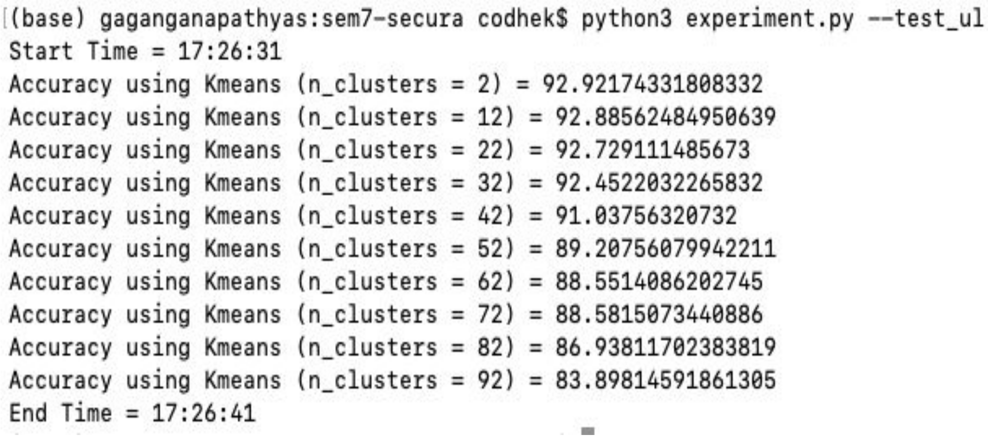
Results

When using only UNSUPERVISED LEARNING



```
(base) gaganganapathyas:sem7-secura codhek$ python3 experiment.py --test_ul
Start Time = 17:04:09
Read file - 50.csv
Read file - 51.csv
Read Dataset...
Total tuples to choose test set from: 389189 = 97850 + 291339
Total non-bot tuples: 48925 -> 14.378541367879059 %
Total bot tuples: 291339 -> 85.62145863212095 %
Total size of test set: 340264
Built testing dataset...
{1: 10, 0: 16602}
Graph built!
16612
Feature Extraction done!
Normalizing Done!
Done pre-processing on Test set!
Accuracy using just Unsupervised Learning = 82.12232121358055
End Time = 17:10:24
```

DBSCAN



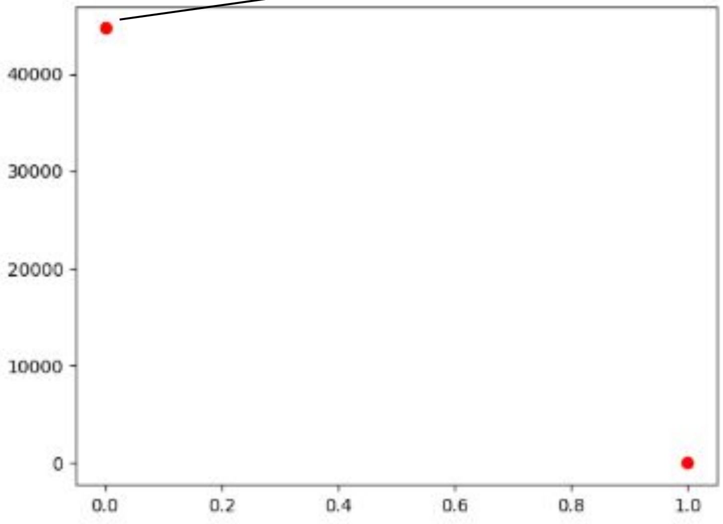
```
(base) gaganganapathyas:sem7-secura codhek$ python3 experiment.py --test_ul
Start Time = 17:26:31
Accuracy using Kmeans (n_clusters = 2) = 92.92174331808332
Accuracy using Kmeans (n_clusters = 12) = 92.88562484950639
Accuracy using Kmeans (n_clusters = 22) = 92.729111485673
Accuracy using Kmeans (n_clusters = 32) = 92.4522032265832
Accuracy using Kmeans (n_clusters = 42) = 91.03756320732
Accuracy using Kmeans (n_clusters = 52) = 89.20756079942211
Accuracy using Kmeans (n_clusters = 62) = 88.5514086202745
Accuracy using Kmeans (n_clusters = 72) = 88.5815073440886
Accuracy using Kmeans (n_clusters = 82) = 86.93811702383819
Accuracy using Kmeans (n_clusters = 92) = 83.89814591861305
End Time = 17:26:41
```

K-MEANS

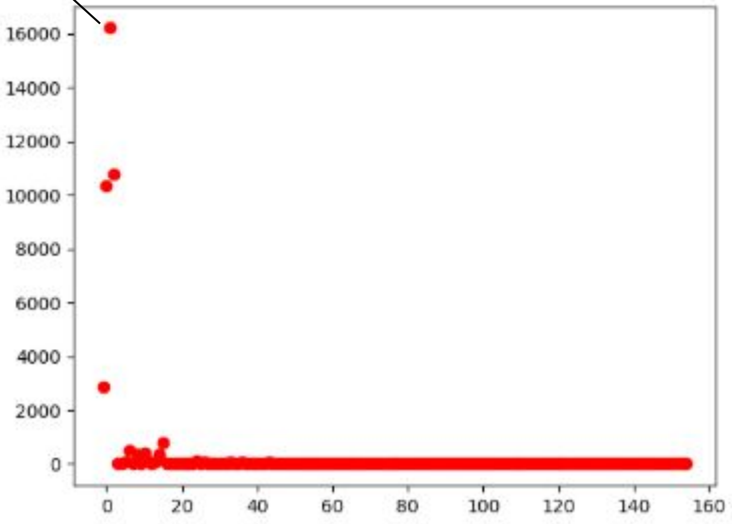
Results

PHASE 1: (Unsupervised Learning)

Non-bot cluster



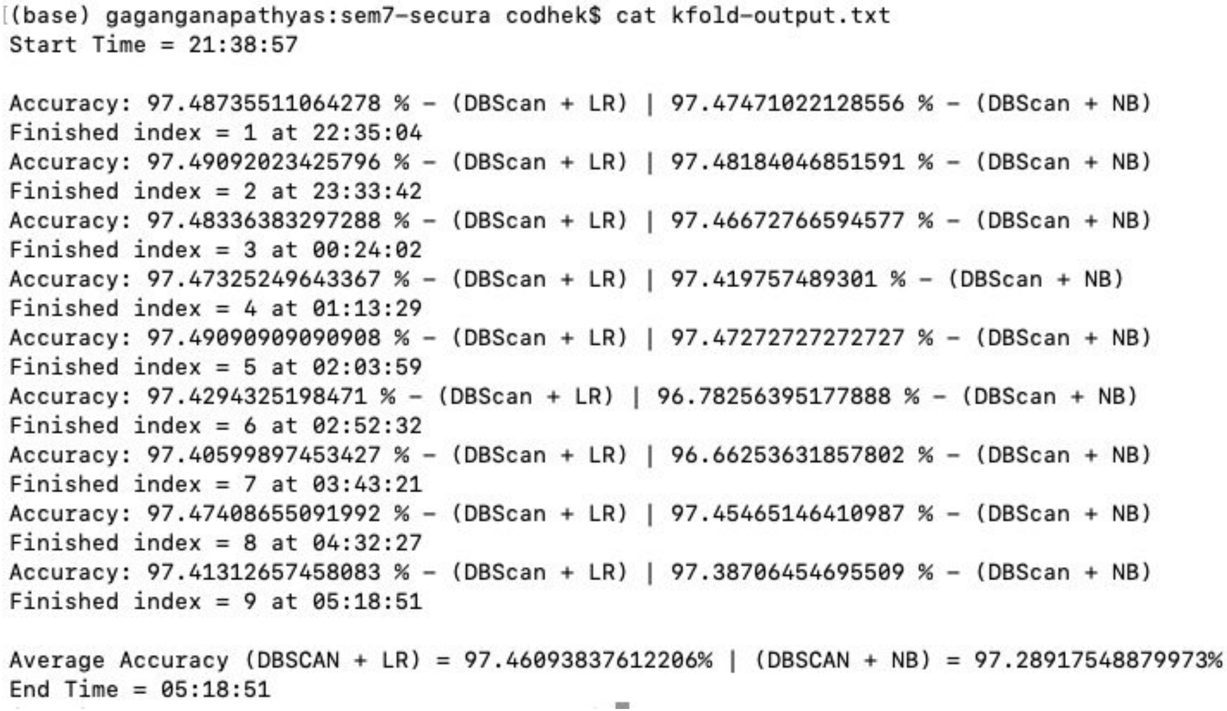
Kmeans (n_clusters=2, random_state=0)



DBScan (eps=0.4, min_samples=4)

Results

Using K-fold cross-validation on all 9 datasets :



```
(base) gaganganapathyas:sem7-secura codhek$ cat kfold-output.txt
Start Time = 21:38:57

Accuracy: 97.48735511064278 % - (DBScan + LR) | 97.47471022128556 % - (DBScan + NB)
Finished index = 1 at 22:35:04
Accuracy: 97.49092023425796 % - (DBScan + LR) | 97.48184046851591 % - (DBScan + NB)
Finished index = 2 at 23:33:42
Accuracy: 97.48336383297288 % - (DBScan + LR) | 97.46672766594577 % - (DBScan + NB)
Finished index = 3 at 00:24:02
Accuracy: 97.47325249643367 % - (DBScan + LR) | 97.419757489301 % - (DBScan + NB)
Finished index = 4 at 01:13:29
Accuracy: 97.49090909090908 % - (DBScan + LR) | 97.47272727272727 % - (DBScan + NB)
Finished index = 5 at 02:03:59
Accuracy: 97.4294325198471 % - (DBScan + LR) | 96.78256395177888 % - (DBScan + NB)
Finished index = 6 at 02:52:32
Accuracy: 97.40599897453427 % - (DBScan + LR) | 96.66253631857802 % - (DBScan + NB)
Finished index = 7 at 03:43:21
Accuracy: 97.47408655091992 % - (DBScan + LR) | 97.45465146410987 % - (DBScan + NB)
Finished index = 8 at 04:32:27
Accuracy: 97.41312657458083 % - (DBScan + LR) | 97.38706454695509 % - (DBScan + NB)
Finished index = 9 at 05:18:51

Average Accuracy (DBSCAN + LR) = 97.46093837612206% | (DBSCAN + NB) = 97.28917548879973%
End Time = 05:18:51
```

Conclusion

Calculating the average over all the datasets we get **97.46%** using Logistic Regression with DBSCAN and **97.29%** using Naive Bayes with DBSCAN. Therefore, our 2 phase learning technique out performs the vanilla single phase learning technique by more than **10%.**

References

https://www.f5.com/services/resources/glossary/bot-mitigation

https://www.stratosphereips.org/datasets-ctu13

BotGM: Unsupervised Graph Mining to Detect Botnets in Traffic Flows - Sofiane Lagraa , Jerome Francois, Abdelkader Lahmadi , Marine Miner,, Christian Hammerschmidt and Radu State.

R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi,N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "**A comprehensive survey on machine learning for networking: evolution, applications and research opportunities**," Journal of Internet Services and Applications, vol. 9, no. 1, pp. 1–99, 2018

"**Botnet detection using graph-based feature clustering**", Sudipta Chowdhury, Mojtaba Khanzadeh , Ravi Akula , Fangyan Zhang,, Song Zhang , Hugh Medal , Mohammad Marufuzzaman and Linkan Bian, 2017

Thank you!