
milcom
Military Communications for the 21st Century
November 12–14, 2019 • Norfolk, VA, USA
Defining Multi-Domain Command and Control

milcom
Military Communications for the 21st Century
November 12–14, 2019 • Norfolk, VA, USA
Defining Multi-Domain Command and Control

Domain Flux-based DGA Botnet Detection Using Feedforward Neural Network

Md. Ishtiaq Ashiq Khan, Protick Bhowmick,
Md. Shohrab Hossain, and **Husnu S. Narman**

Outlines

- Motivation
- Problem
- Contribution
- Results
- Conclusions

Identifying Jargons

Domain Flux -based DGA Botnet Detection Through Feedforward Neural Network

- BOTNET
- DOMAIN FLUX
- DGA
- FEEDFORWARD NEURAL NETWORK

Motivation

- Military communication involves the transmission of heavily secured information.
- Even a minor infiltration of military network can be catastrophic.
- One way of invading into this network is botnet.

Problem

- Botnets Detections
  - Domain fluxing method, in which botmaster constantly changes the domain name of the Command and Control (C&C) server very frequently.
  - These domains are produced using an algorithm called Domain Generation Algorithm (DGA).
  - Domain flux-based botnets are stealthier and consequently much harder to detect due to its flexibility.

Some Solutions and Limitations

- Not well-formed and pronounceable domain names
- Identify differences between human-generated domains and DGAs
- Detecting malicious domain names by comparing its semantic similarity with known malicious domain names
- Domain length which could be different from domain name

- Fail: Random meaningful word phrases
- Fail: DGA domains showing a bit of regularity

Contributions

- Developed a heuristic for evaluation and detection of botnets inspecting the several attributes in a very simple and efficient way
- Compared our proposed system with the existing ones with respect to accuracy, F1 score, and ROC curve

Proposed Features

- Length
- Vowel-consonant ratio
- Four-gram Score
- Meaning Score
- Frequency Score
- Correlation Score
- Markov Score
- Regularity Score

Length & Vowel -consonant ratio

Domain Name	Length	Vowel-consonant ratio	Comment
aliexpress	10	0.667	Normal
xxtrlasffbon	12	0.2	Abnormally low ratio
aliismynameexpress	19	0.55	Abnormal length

Four-gram Score

Domain Name	No. of four-grams without a vowel	Comment
google	0	Normal
xxtrlasffbon	3 (xxtr, xtrl, sffb)	Abnormal but detectable by v-c ratio (0.2)
bbxtklaoeo	3 (bbxt, bxtk, xtkl)	Abnormal and not detectable by v-c ratio (0.667)

Regularity Score

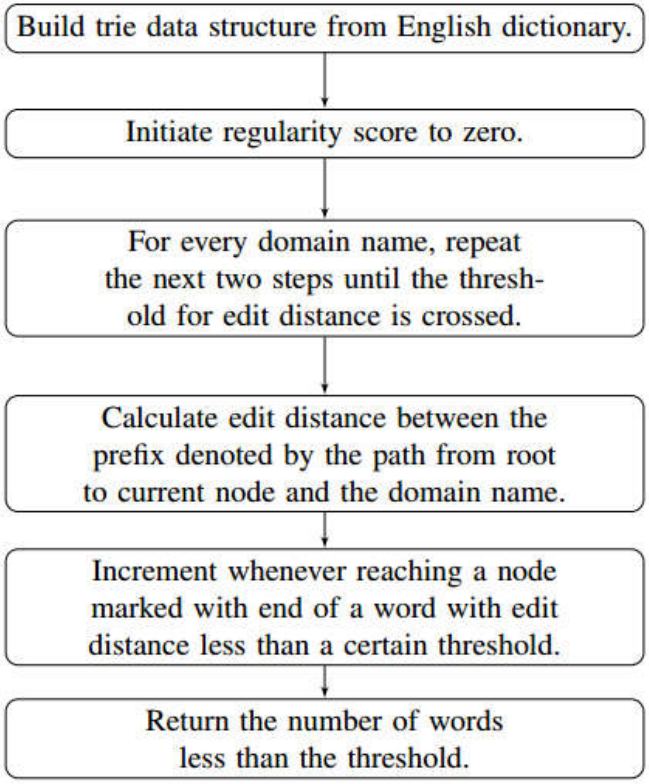


Fig. 1: Regularity Score step by step

- The regularity score takes into account the syntactic dissimilarity with actual words by using Edit distance.
- Edit distance takes two words as function parameters and returns the minimum number of deletions, insertions, or replacements to transform one word into another.

Regularity Score: Example

- Let's build a "trie" from two words "coco" and "coke"
- Let's say, our threshold is 1.
- c → o → c → o
  ↓
  k → e
- Let the domain names be "coca" and "caket"
- For "coca", similarity score will be 1 -> (threshold is 1, coco)
- For "caket", similarity score will be 0 -> (threshold is 1, N/A)

So, Regularity Score of caket > coca
So, DGA probability (caket > coca)

Markov Score

- A big text file was chosen to build the Markov model.
- Every transition between adjacent letters were taken into account to calculate the transition probability.
- A 2-D array was used to store the transition frequencies, and afterwards the values were normalized to find the transition probabilities.
- In training phase, for every 2-grams within a domain name, the sum of the transition probabilities were calculated to generate the score.

Markov Score: Example

- Let's say the training text consists of a single word "begone" and the test set is "banet" and "nebet"
- So, the transition matrix will be:

  t[b][e] = 1, t[e][g] = 1, t[g][o] = 1, t[o][n] = 1, t[n][e] = 1

- For "banet", t[b][a] + t[a][n] + t[n][e] + t[e][t] = 0 + 0 + 1 + 0 = 1
- For "nebet", t[n][e] + t[e][b] + t[b][e] + t[e][t] = 1 + 0 + 1 + 0 = 2

So, Markov Score of nebet > banet
So, DGA probability (banet > nebet)

Meaning Score

- Basis:
  - Real world domain names tend to include meaningful words or phrases.
- Methodology:
  - Meaningful segments extracted from a domain name
  - Normalized with respect to length

Meaning Score: Example

peerscale

1. Meaningful substrings (peer, scale)
2. Two of length 4 & 5

ononblip

1. Meaningful substrings (blip)
2. Only 1 of length 4

Overall, Meaning Score of ononblip < peerscale
So, DGA probability (ononblip > peerscale)

Frequency Score

- Depends on the relative use of the word over the internet
- Steps:
  1. Substrings of length greater than three extracted from the domain names in the training set
  2. Relative frequency of the substrings determined from Google Books N-gram dataset
  3. Score generated from the relative frequency of the substrings scaled exponentially by the length of substrings

Frequency Score: Example

peerscale

1. Extracting substring of length greater than three (ersc, eers, peer, scale etc.)
2. Sorted according to frequency score (ersc < eers < peer < scale)

ononblip

1. Extracting substring of length greater than three (onon, blip, nbli, nonb etc.)
2. Sorted according to frequency score (nbli < nonb < onon < blip)

Overall, Frequency Score of ononblip << peerscale
So, DGA probability (ononblip > peerscale)

Correlation Score

- Depends on whether the word segments in the domain have a contextual similarity
- Steps:
  1. Extract lines from the reference text file
  2. Update correlation map for every pair of words within a sentence
  3. Extract substrings from the domain names in the training set
  4. Check the incidence of the substrings appearing together from our correlation map
  5. Generate correlation score based on substring length and prevalence

Correlation Score: Example

- Let's say the reference text consists of a single line "I hate menial work" and the domains in question are "workhaters" and "clustolous"
- So, the correlation map will be:

  c[I][hate] = 1, c[I][menial] = 1, c[I][work] = 1, c[hate][menial] = 1, c[hate][work] = 1, c[menial][work] = 1
- For "workhaters", correlation score is 1
- For "clustolous", correlation score is 0.

So, Correlation Score of workhaters > clustolous
So, DGA probability (clustolous > workhaters)

Results

- Experiment
- Dataset
- Used performance metric
  - Accuracy
  - F1 Score
  - ROC (Receiver operating characteristic) Curve and AUC (Area Under the ROC curve)
- Results

Dataset

- We collected our data set from the research work of F. Yu. et al.
- Three folders
  - hmm_dga : domains generated using Hidden Markov model
  - pcfg_dga: domains generated using Probabilistic Context Free Grammar
  - other: some real world known botnet domains

Performance Metric

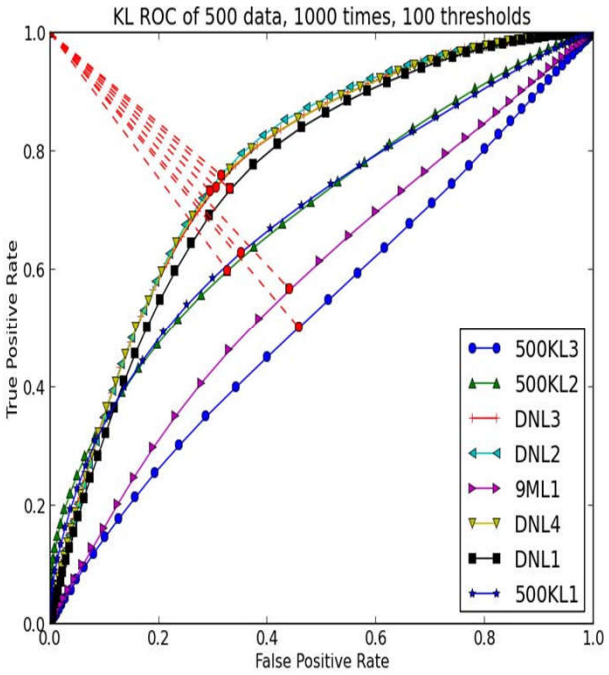
File Name	Test Accuracy	F1 score	AUC score	Comment
9ML1	92%	0.92	0.96	Excellent
500KL1	96%	0.96	0.98	Excellent
500KL2	95%	0.95	0.98	Excellent
500KL3	86%	0.86	0.93	Excellent
DNL1	85%	0.85	0.92	Excellent
DNL2	82%	0.82	0.89	Excellent
DNL3	81%	0.81	0.88	Good
DNL4	81%	0.81	0.88	Good
kraken	96%	0.96	0.99	Excellent
srizbi	97%	0.97	0.98	Excellent
torpig	99%	0.99	0.99	Excellent
zeus	100%	1.00	1.00	Excellent
conflicker	97%	0.97	0.98	Excellent
kwyjibo	70%	0.70	0.79	Moderate
pcfg_dict	70%	0.70	0.77	Moderate
pcfg_dict_num	73%	0.73	0.79	Moderate
pcfg_ipv4	85%	0.85	0.92	Excellent
pcfg_ipv4_num	86%	0.86	0.94	Excellent

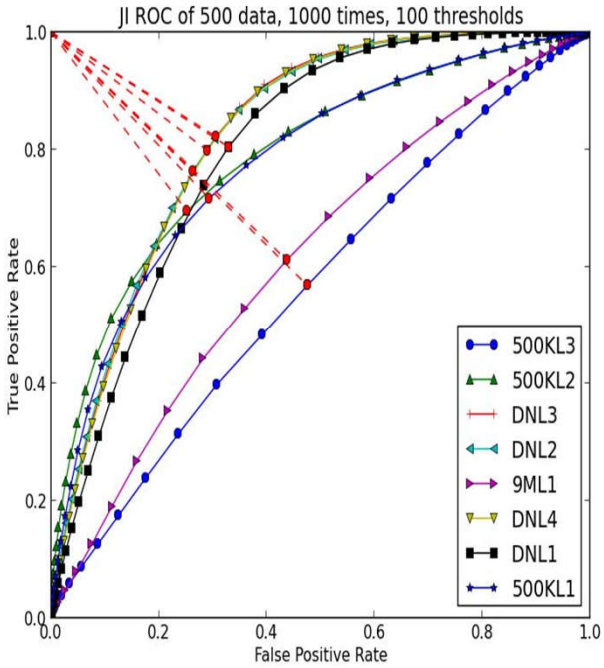
If AUC score is greater than 0.9, we call it *excellent*. If it falls within the range 0.80-0.9, it is *good*. Within 0.70-0.80 is *moderate* and anything less than 0.70 is termed as *poor*.

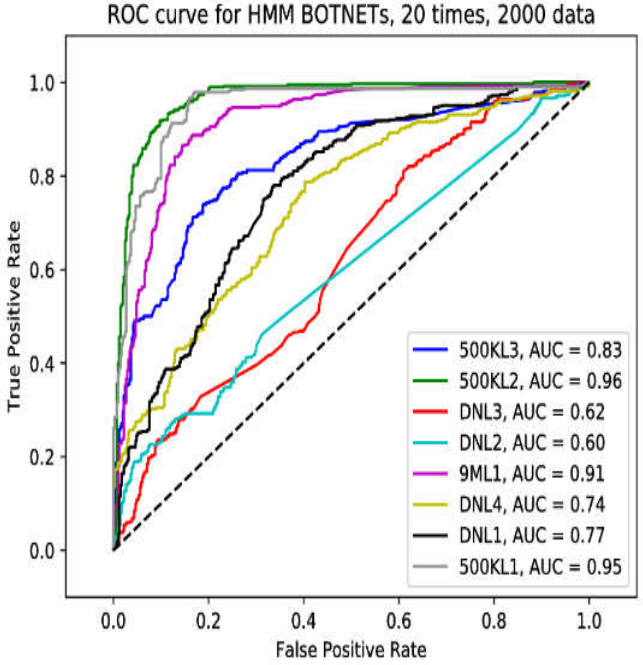
Our Results

- Our baseline approach is the method proposed by S. Yadav et. Al.
- They proposed three metrics to determine DGA domain
  - KL (Kullback-Leibler) distance
  - Jaccard Index
  - Edit Distance

Our Results: Graphical Comparison

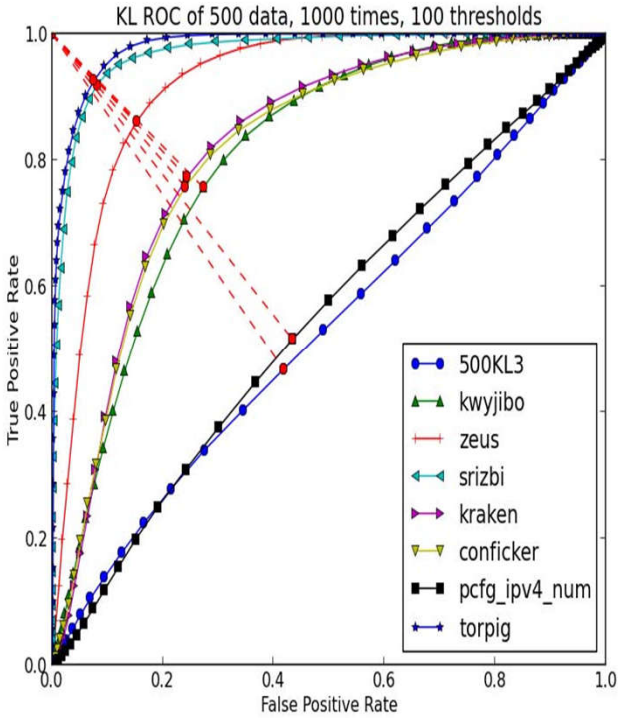


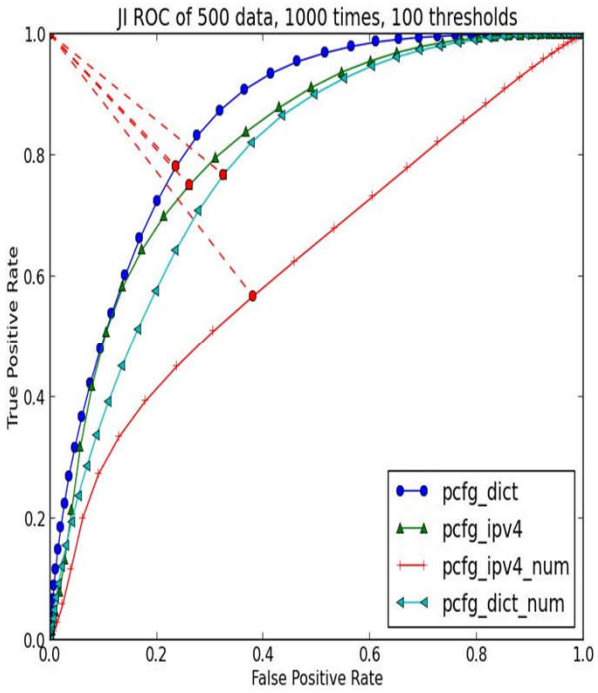


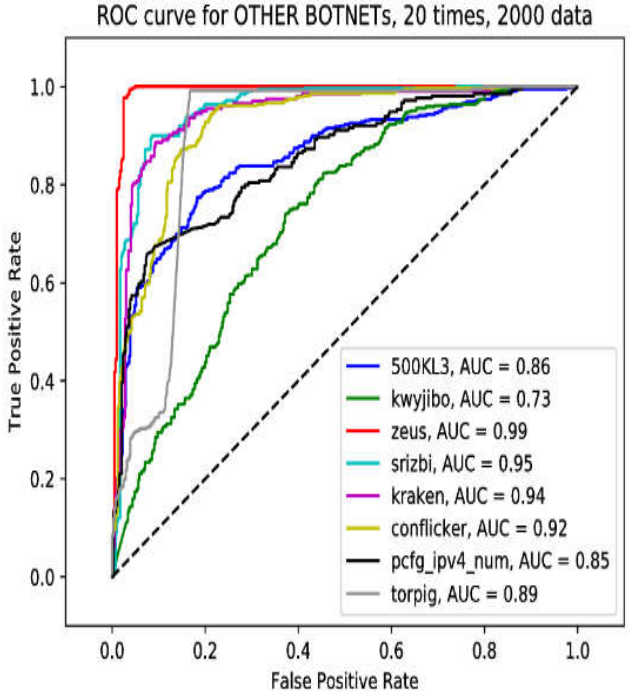


For ‘hmm_dga’ folder

Our Results: Graphical Comparison

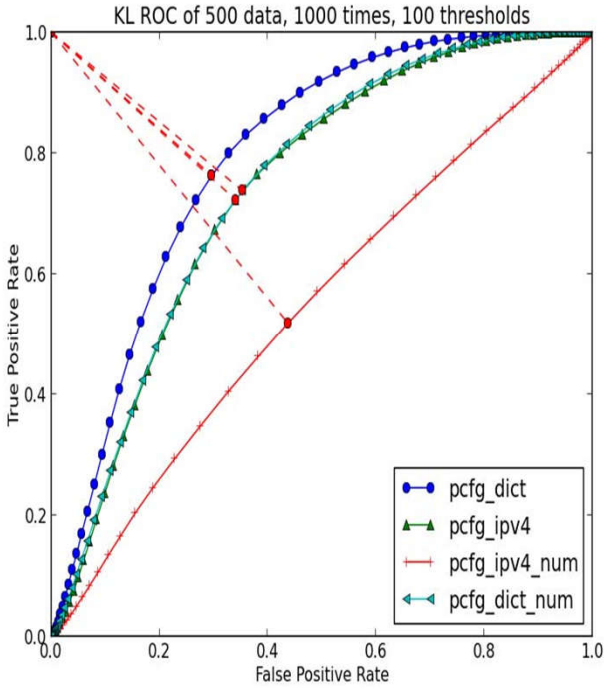


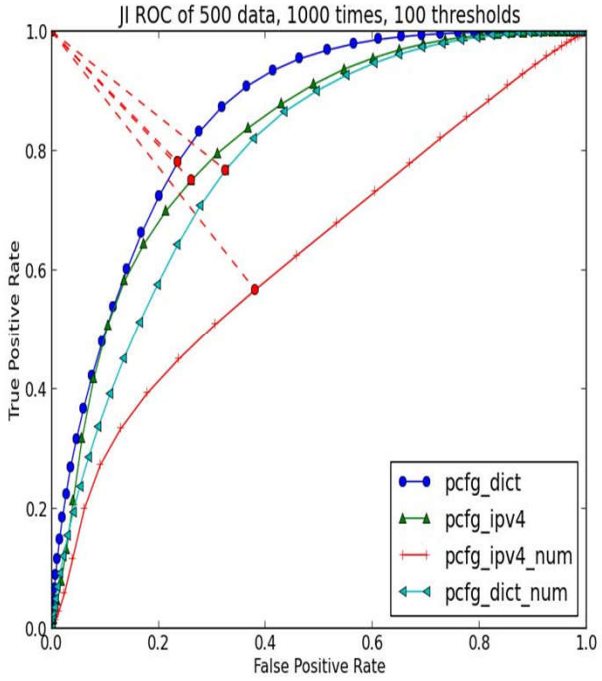


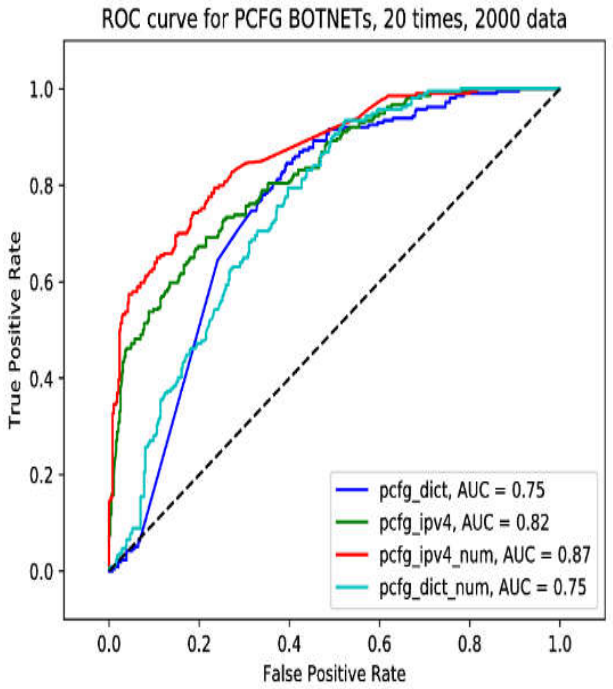


For ‘other’ folder

Our Results: Graphical Comparison







For ‘pcfg_dga’ folder

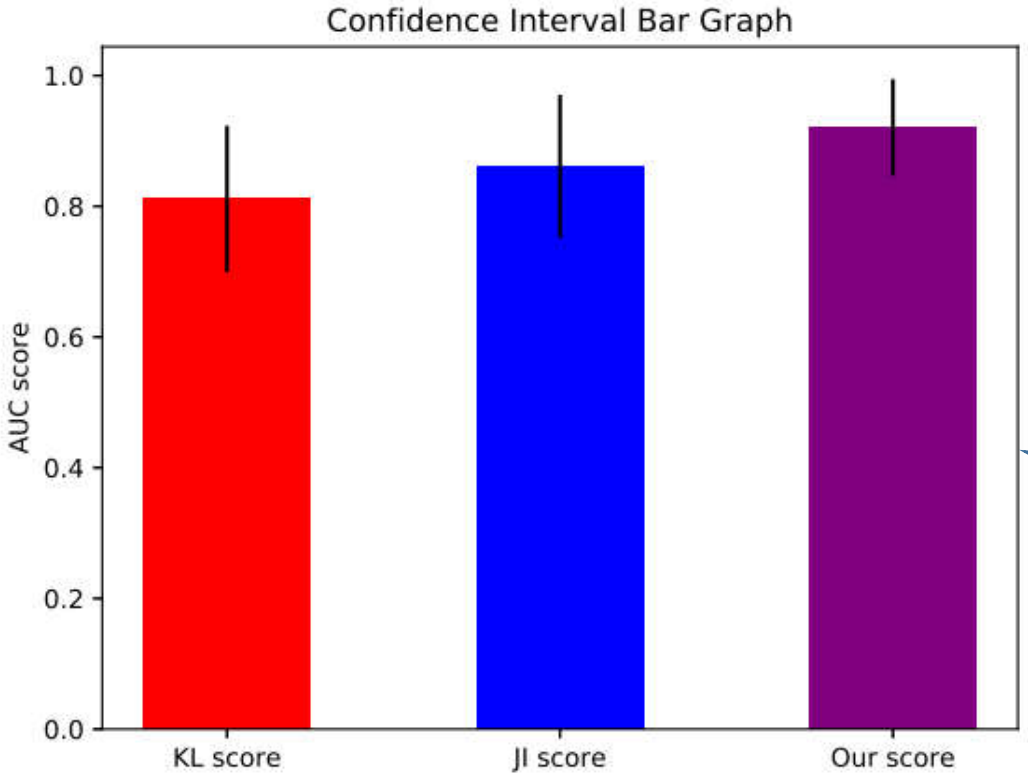
Our Results: Quantitative Comparison

File	KL score	JI score	Our result
9ML1	0.70	0.70	0.96
500KL1	0.84	0.90	0.98
500KL2	0.86	0.92	0.98
500KL3	0.55	0.62	0.93
DNL1	0.84	0.91	0.92
DNL2	0.83	0.90	0.89
DNL3	0.84	0.89	0.88
DNL4	0.84	0.87	0.88
kraken	0.88	0.90	0.99
srizbi	0.95	0.91	0.98
torpig	0.95	0.99	0.99
zeus	0.94	1.00	1.00
conflicker	0.89	0.88	0.98
kwyjibo	0.81	0.89	0.79
pcfg_dict	0.80	0.88	0.77
pcfg_dict_num	0.75	0.86	0.79
pcfg_ipv4	0.75	0.88	0.92
pcfg_ipv4_num	0.58	0.60	0.94

Well detecting HMM-based and real IP domains.

Not better than KL or JI for pronounceable words

Our Result: Confidence Interval Bar Graph



The confidence interval suggests that variation of result in our system are not be as much as the other two methods.

Our Result: Key Findings

- For files containing numbers, our approach seems to be better than the reference.
- For files containing domains from real life botnets, our approach produced much better result.
- For files with pronounceable domains, results of baseline approach is slightly better than ours.

Conclusion

- Our system considers the problem from two aspects - syntactically and semantically.
- The result is exceptionally well on DGAs that use pseudo random number generator.
- Frequency Score and Meaning Score are good classifiers for DGAs that use pronounceable domain names.
- When related phrases and words appear within the domain names, value of correlation score is a good classifier.

FUTURE WORKS

- Incorporate more semantic features in future

Thank You
Questions

Husnu Narman
narman@marshall.edu
https://hsnarman.github.io/