Detecting Word Based DGA Domains Using Ensemble Models

P.V. Sai Charan, Sandeep K Shukla and P. Mohan Anand
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, India

Presenter : P. V. Sai Charan
Email: pvcharan@cse.iitk.ac.in

Date : 14-12-2020
Conference : CANS 2020

Agenda

1. Introduction
2. Brief History about DGA families
3. Issues with current approaches (Literature survey)
4. Proposed Methodology
5. Experimental Results & Analysis
6. Future work
7. Summary

Introduction

- Modern-day malware are intelligent enough in evading detection of Control and Command server (C2C) infrastructure by using various advanced techniques.
- Domain Generation Algorithms (DGA) is one such popular evasive technique to contact C2C [1]
- Usage is rapidly increasing in Advanced persistent Threat (APT), Ransomware & Botnet attacks in recent times [2]

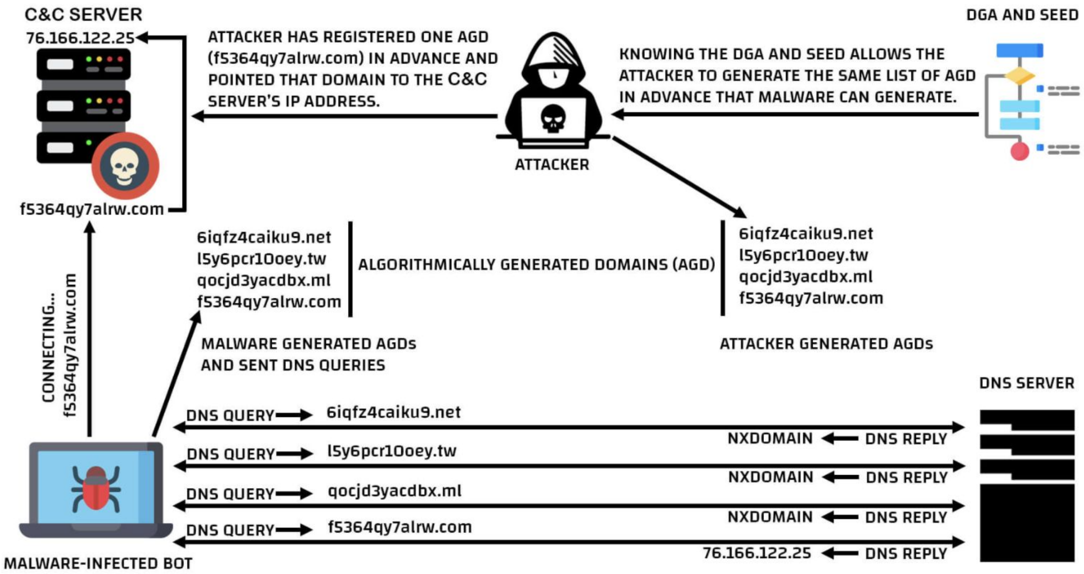


Fig.1. DGA domains in attack scenario [3]

Brief History of DGA Domains

1. Legacy Malware developers used to hard code the IP address of C2C in malware payload itself

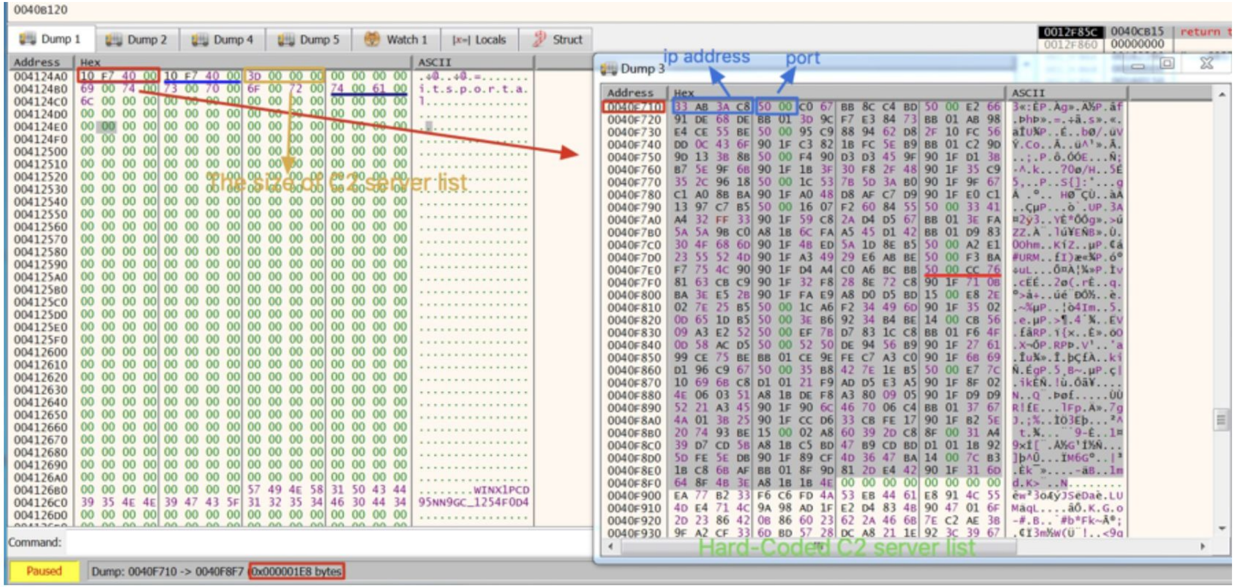


Fig.2. Hardcoded C2C list in emotet malware [4]

Catch : Hardcoded IP address can be simply found out during reverse engineering of malware payload

Brief History of DGA Domains

2. Attackers generate a list of domains using Pseudo Random Number Generators (PRNG's)

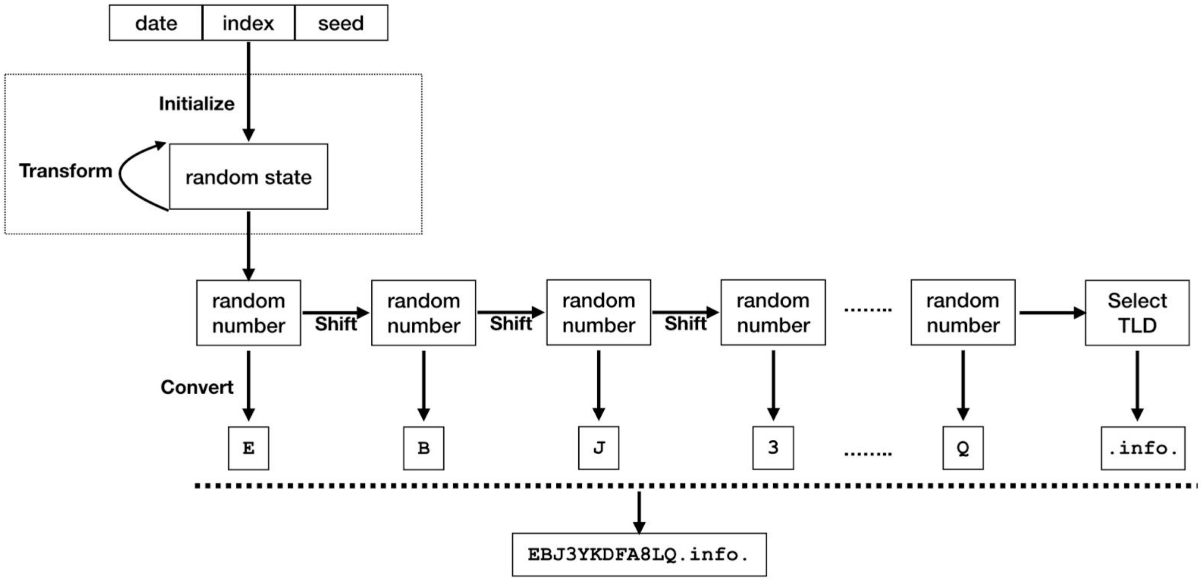


Fig.3. Character Based DGA-PRNG [5]

Note : Recent Advances in malware research addressed this problem to a large extent [6]

Brief History of DGA Domains

3. word based DGA - malware writers uses a set of words from dictionary to construct meaningful substrings that resembles real domain names.

Example : crossmentioncare.com , manygoodnews.com

- **Matsnu** - Contains 2 to 3 words from a preferred dictionary and can generate 10 domains per day. [com] is the possible TLD. (world-bite-care.com, activitypossess.com, mattermiss-type.com)
- **SuppoBox** - Contain [net,ru] as TLD Combines two words from the word lists. Can generate 254 domains per day. (tablethirteen.net childrencatch.net)
- **Gozi** - Widely used in banking trojans and rootkits that persist for a long time in sensitive corporate networks (morelikestoday.com, sociallyvital.com)

Pzid, CryptoWall, Volatile, Banjori are other families of Word based DGA Malware.

Issues with Word Based DGA Detection

Key Issue : **Proximity to Real world domains**

- **Plohmann et.al - Comprehensive study on DGA malware [7]**
  - Explains Complexity of Word-list based DGA families and their detection
- **Curtin et.al - Detecting domains with recurrent neural network [8]**
  - **Smashword Score** (measures how much DGA domain is close to the English word)
  - Issue : Not adaptable for corporate use (Matsnu - 89% , Gozi-77.3%, Suppobox-79.8%)
- **Luhui et.al - Detecting wordsbased DGA using semantic Analysis [9]**
  - Front-word-correlation (FWC) & Back-word-correlation (BWC)
  - Issue : Poor Accuracy (~0.83) with High False positives

Issues with Word Based DGA Detection

- **Woodbridge et.al - Predicting wordbased Domains using LSTM neural network [10]**
  - Needs no feature extraction & less classification time
  - Issue : Class imbalance ; Failed to detect Suppobox and Matsnu families
- **Jasper et.al - DGA detection using popularity method [11]**
  - Sudden increase to traffic flow to a particular is monitored over the period of time
  - Issue : Minimum 1 day to observe changes in network; Not suitable for realtime
- **Choi et.al - BotGAD framework to detect malicious domain [12]**
  - Captures all DNS traffic passing through the network.
  - Issues : Depends only on TTL records ; Easily evaded by modern APT ‘s & Botnets

Proposed Model

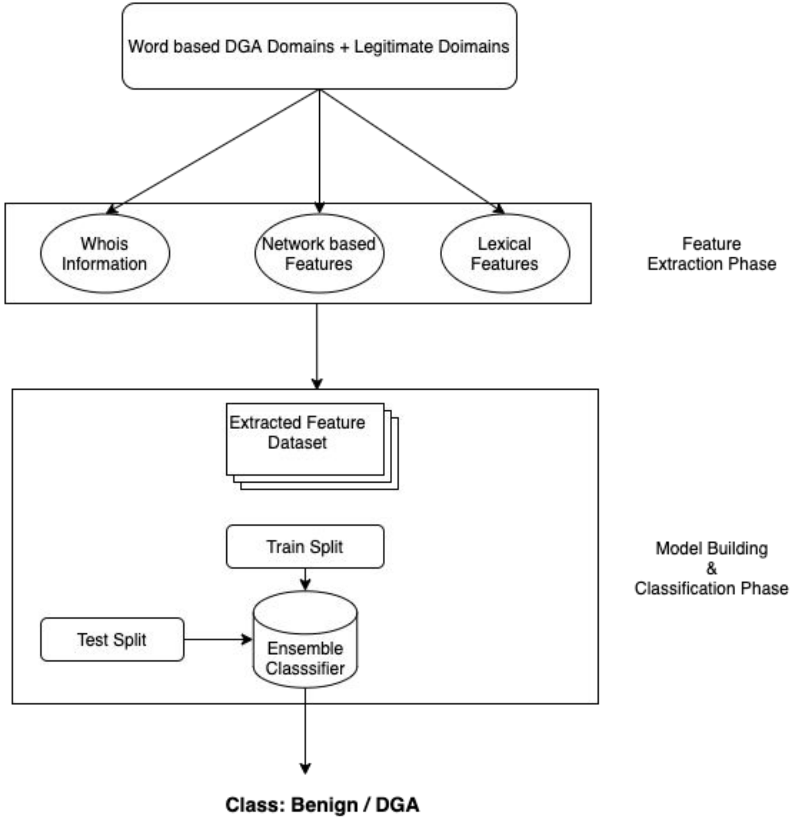


Fig.4. Proposed Model for Word Based DGA detection

S.NO	Feature	Example(crossmentioncare.com)
1	Domain Name	crossmentioncare.com
2	Word Count	3
3	Length	16
4	Syllable Count	4
5	Vowel Count	6
6	Consonant Count	10
7	Created Since(in days)	2192
8	Updated Since(in days)	2189
9	Registrar(Binary)	1
10	TTL (in seconds)	86400
11	IANA (Binary)	1
12	Unique Letters	10
13	Hyphen (Binary)	0
14	Underscore (Binary)	0
15	Family Type	MATSNU

Table 1. Features considered for MATSNU domain

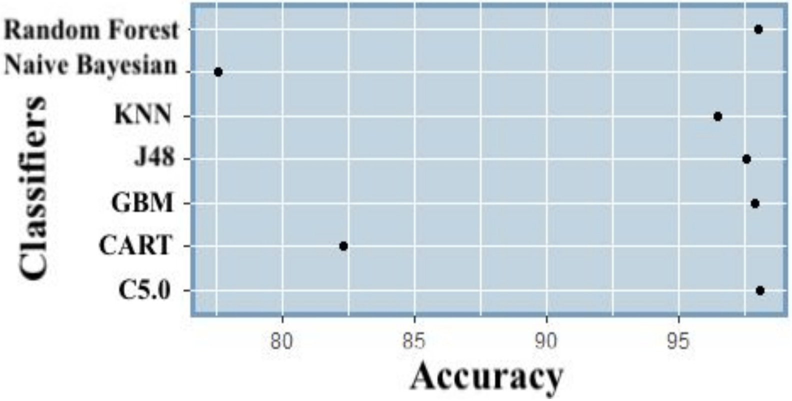
Experiment Results & Analysis

GOAL : We performed 5 experiments to reduce feature set and improve accuracy

1. 15 Features for model training + Feature Correlation Analysis.
2. Top 8 Features for model training (from Feature Importance Analysis)
3. Principal Component Analysis on 15 feature dataset (Linear Dimensionality reduction technique)[13]
4. Diffusion Map on 15 feature dataset (Non-linear Dimensionality reduction technique) [14]
5. Robustness Analysis of our model (Synthetic data generated using CTGAN [15])

Experiment - 1

- Considered all 15 features for constructing model training
- 40000 samples (10000 random samples from each class i.e Matsnu, Suppobox, Gozi, Bening)



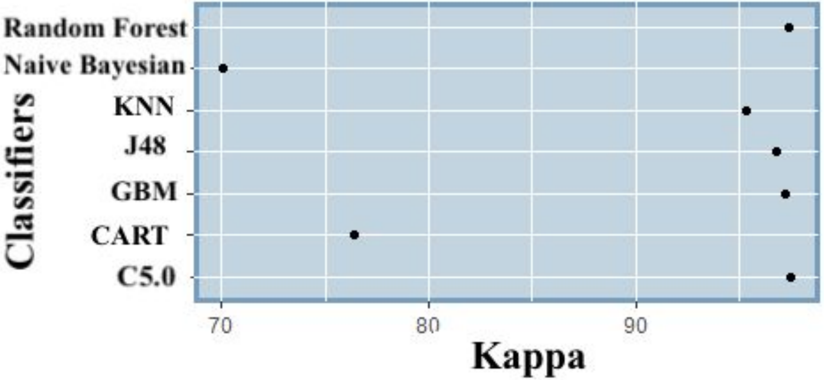


Fig.5. Accuracy and Kappa Graph for various classifiers for 15 feature dataset

Take Away : **C5.0 Stands out to be Best Performer (Low FPR + Low FNR + Low Training time)**

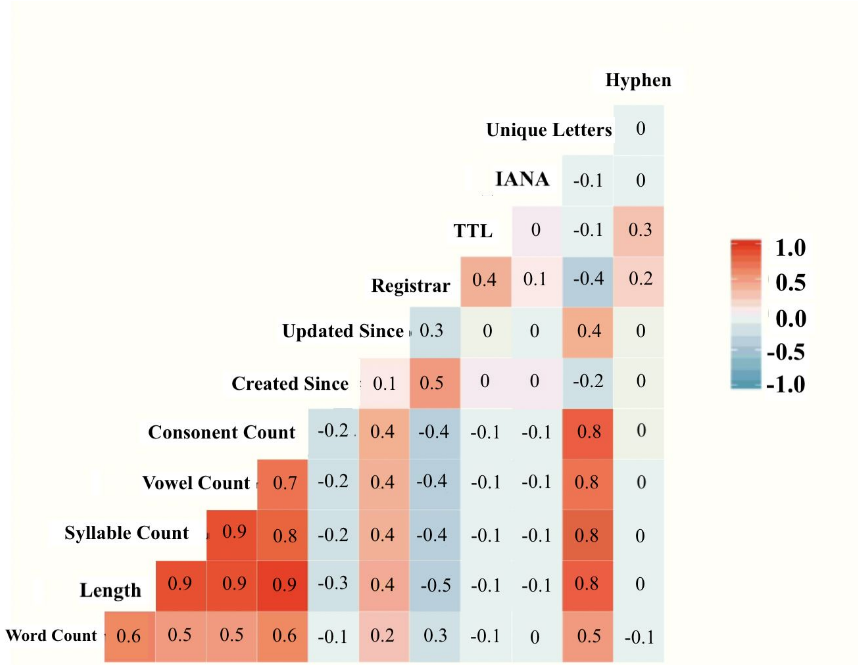


Fig.6. Feature Correlation Analysis for 15 feature dataset

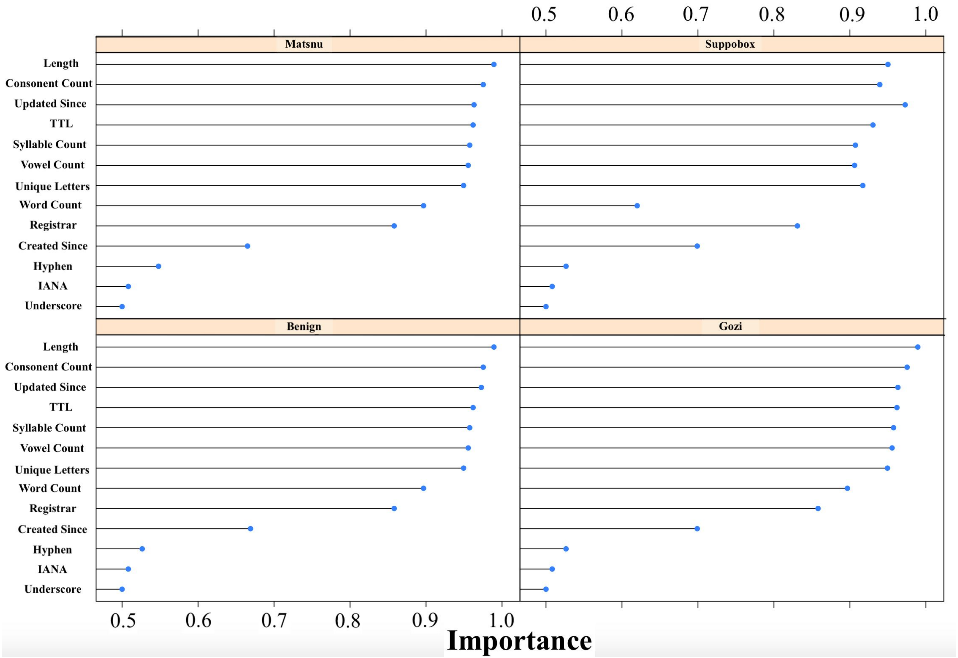


Fig.7. Feature importance Graph for 15 feature dataset

Experiment - 2

- We consider top 8 features to train our model (4 - Lexical + 4 - Network based)
- We achieve almost similar accuracy (2% drop) by reducing half of features

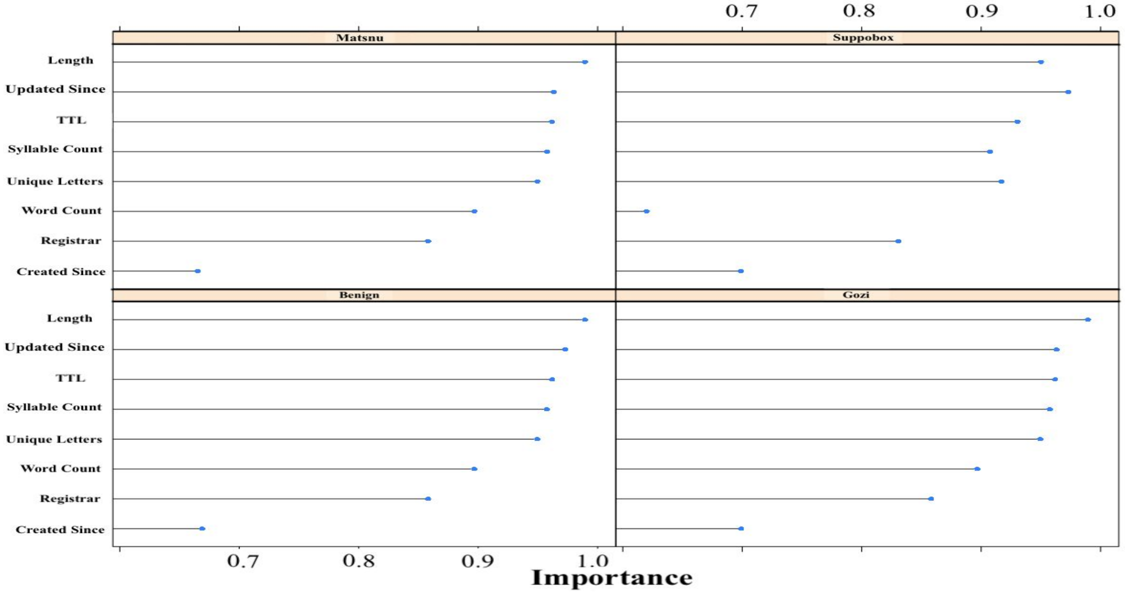


Fig.8. Feature Importance for 8 feature dataset

Take Away : Random Forest tops in terms of accuracy but it's training time and model size is almost double than C5.0

Experiment - 3

- We apply Principal Component Analysis on 15 feature dataset.
- Our observation , 4 % drop in accuracy by considering **top 8** Principal Components

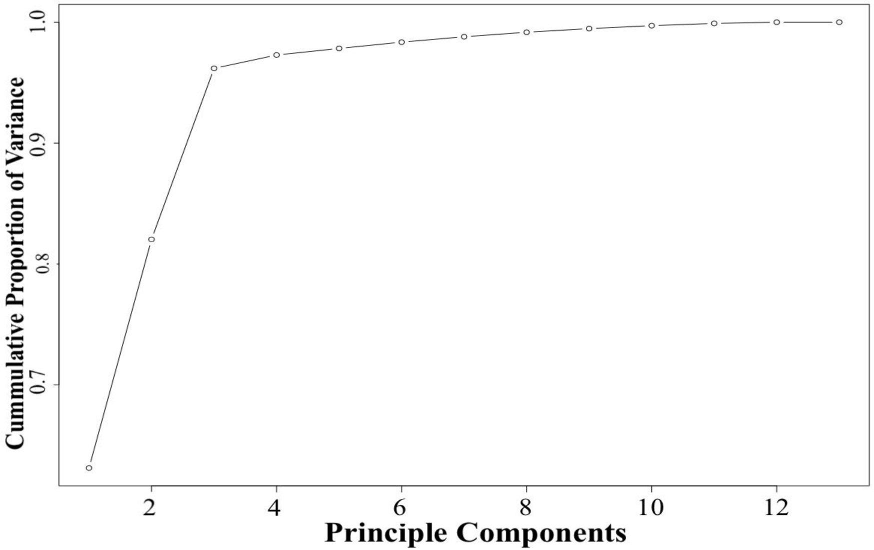


Fig.9. Principal Components vs Variance plot

Take Away : We observe a large number of GOZI, MATSNU, SUPPOBOX families misclassified as benign i.e less significant principal components are impacting decision stumps of ensemble models.

Experiment - 4

- We apply Diffusion map on 4800 samples (1200 sample from each type)
- In addition we applied K-means on normal space & Diffusion space

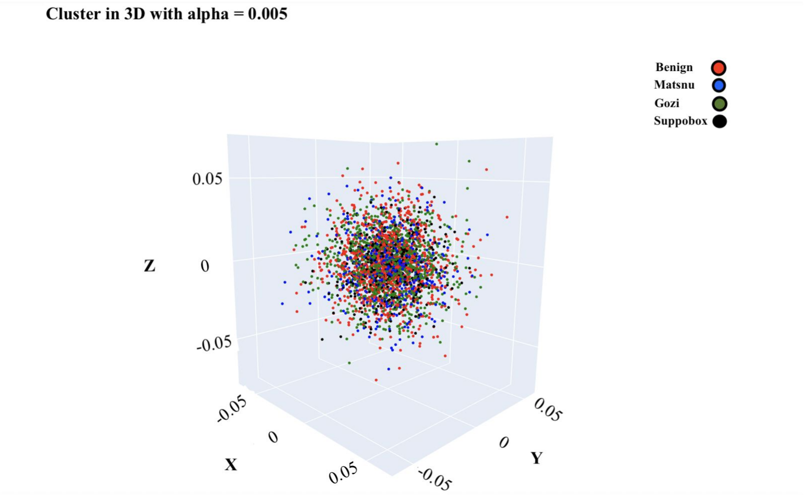


Fig.10. Diffusion Map with alpha=0.005

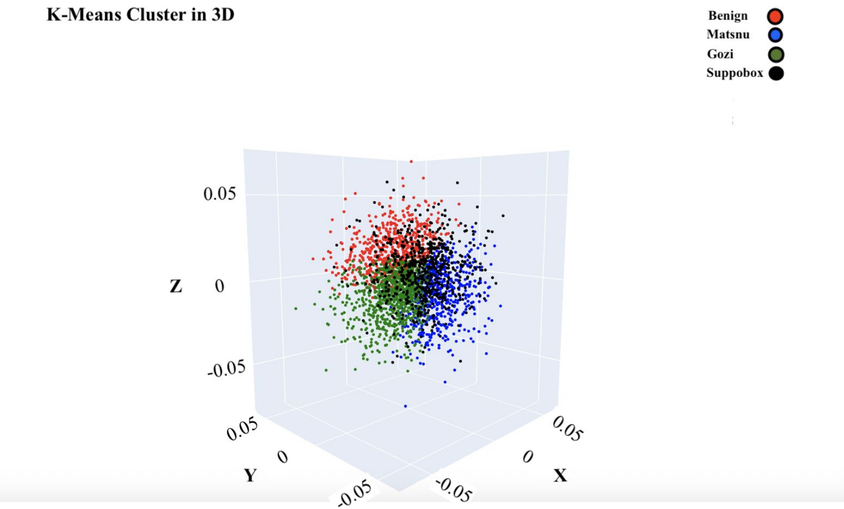


Fig.11. K-means on Diffusion Map data (alpha=0.005)

Take Away : There is no underlying structure for this dataset

Experiment - 5

- We test Robustness of our model in this experiment using CTGAN
- Tested our model with 30000 synthetic data samples (10000 from each DGA family) + 4000 legitimate.

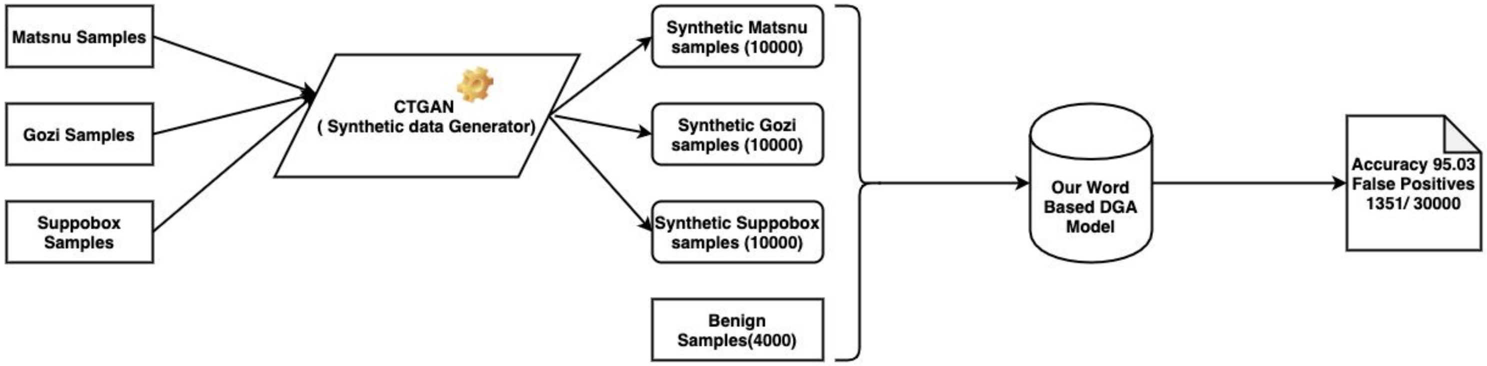


Fig.12. Generating synthetic data for DGA families using CTGAN

Take Away : Our model did a decent work by classifying malicious and benign domains with 0.9503 Accuracy

Summary & Future Scope

In this paper, we mainly addressed :

1. **Ensemble models for detecting word based DGA families (GOZI, MATSNU, SUPPOBOX)**
2. **Linear & Nonlinear dimensionality reduction techniques to understand underlying structure of data**
3. **CTGAN to generate synthetic test data to verify robustness of our models**

Possible Future works :

- **Extend this approach for emerging DGA families**
- **GAN to generate synthetic data for future DGA families → Building robust botnet/malware models**

References

1. Chen, Xu, et al.: Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware. In: IEEE international conference on dependable systems and networks with FTCS and DCC (DSN), pp. 177-186. IEEE (2008).
2. Charan, PV Sai, T. Gireesh Kumar, and P. Mohan Anand.: Advance Persistent Threat Detection Using Long Short Term Memory (LSTM) Neural Networks. In: International Conference on Emerging Technologies in Computer Engineering, pp. 45-54. Springer, Singapore (2019).
3. DGA in Malware; https://hackersterminal.com/domain-generation-algorithm-dga-in-malware/
4. Deep dive into emotetr malware : https://www.fortinet.com/blog/threat-research/deep-dive-into-emotet-malware
5. A death match of DGA : https://blogs.akamai.com/2018/01/a-death-match-of-domain-generation-algorithms.html
6. Anand, P. Mohan, T. Gireesh Kumar, and PV Sai Charan.: An Ensemble Approach For Algorithmically Generated Domain Name Detection Using Statistical And Lexical Analysis. Procedia Computer Science 171, 1129-1136 (2020).
7. Plohmann, Daniel, Khaled Yakdan, Michael Klatt, Johannes Bader, and Elmar Gerhards-Padilla.: A comprehensive measurement study of domain generating malware. In: 25th {USENIX} Security Symposium ({USENIX} Security 16), pp. 263-278 (2016).
8. Curtin, Ryan R., Andrew B. Gardner, Slawomir Grzonkowski, Alexey Kleymenov, and Alejandro Mosquera.: Detecting DGA domains with recurrent neural networks and side information. In: Proceedings of the 14th International Conference on Availability, Reliability and Security, pp. 1-10 (2019).

References

9. Yang, L., Zhai, J., Liu, W., Ji, X., Bai, H., Liu, G., Dai, Y.: Detecting word-based algorithmically generated domains using semantic analysis. Symmetry, 11(2), 176 (2019).
10. Woodbridge, Jonathan, Hyrum S. Anderson, Anjum Ahuja, and Daniel Grant.: Predicting domain generation algorithms with long short-term memory networks. arXiv preprint arXiv:1611.00791 (2016).
11. Abbink, Jasper, and Christian Doerr.: Popularity-based detection of domain generation algorithms. In: Proceedings of the 12th International Conference on Availability, Reliability and Security, pp. 1-8 (2017).
12. Choi, Hyunsang, Heejo Lee, and Hyogon Kim.: BotGAD: detecting botnets by capturing group activities in network traffic. In: Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE, pp. 1-8 (2009).
13. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), pp. 37-52 (1987).
14. De la Porte, J., B. M. Herbst, W. Hereman, and S. J. Van Der Walt.: An introduction to diffusion maps. In: Proceedings of the 19th Symposium of the Pattern Recognition Association of South Africa (PRASA 2008), Cape Town, South Africa, pp. 15-25 (2008).
15. Xu, Lei, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni.: Modeling tabular data using conditional gan. In: Advances in Neural Information Processing Systems, pp. 7335-7345 (2019).

Thank You