Cyber Security Experts Association of Nigeria (CSEAN)
Society for Multidisciplinary & Advanced Research Techniques (SMART)
Faculty of Computational Sciences & Informatics - Academic City University College, Accra, Ghana
SMART Scientific Projects & Research Consortium (SMART SPaRC)
Sekinah-Hope Foundation for Female STEM Education
ICT University Foundations USA

Proceedings of the Cyber Secure Nigeria Conference – 2023

Detection of Algorithmically Generated Domain Names using Ensemble Machine Learning Technique

[1]Abdullahi, S.M., [2]Mohammed, A., [3]Ibrahim, R.Y. & [4]Shamsuddeen, A.
[1]Department of Cyber Security, Air Force Institute of Technology (AFIT), Kaduna - Nigeria.
[2]Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria.
[3]Dept of Computer Science Education, Isa Kaita College of Education, Dutsinma-Katsina, Nigeria.
[4]Iya Abubakar Institute of Computing & ICT, ABU - Zaria, Nigeria.
Phone Nos: +2348065309476; +2348067804447; +2347060767397; +2348061615365;
E-mails: [1]samailaa89@gmail.com, [1]smabdullahi@afit.edu.ng; [2]abdullahilwafu@abu.edu.ng;
[3]ummuhaneefel@gmail.com; [4]shamsu2000@yahoo.com

ABSTRACT

Prior to now, cyber attackers use malwares with hard-coded domain names stored in the malware binaries that communicate with a command and control (C&C) servers to launch cyber-attacks on their victim computers. Malware attacks such as botnets and ransomwares are some of the most prevalent forms of these attacks. As soon as a system is infected with a malware (either a botnet or a ransomware), one of the most essential components is to establish a secured communication with the botmaster (i.e., the malware author), through a C&C server. However, with a simple reverse engineering technique, cyber security experts could detect and block these domain names, hence, denying them the ability to communicate with the C&C servers and from receiving further instructions from the botmaster. This led to cyber criminals developing the Domain Generation Algorithm (DGA) technique, which algorithmically generate thousands or more candidate's domain names for communication with the C&C server, thereby obfuscating the domain names of these malwares and making it difficult for cyber security experts to detect or block these domain names. This paper therefore proposes an ensemble machine learning technique for the detection and classification of algorithmically generated domain names (AGDNs) leveraging the combined strength of 4 different machine learning algorithms: Naïve Bayes, SVM, Random Forest and CART. The models were trained twice, first with 4 features and thereafter with 10 features. In order to effectively utilise the result of the predictions, we used a voting-based ensemble approach, where the final classification is decided by the majority vote of the algorithms. Result of the research shows that the Naïve Bayes model performed better than all the other models with an accuracy of 97.54% when trained with 10 features and 95.99% when trained with 4 features.

Keywords: WSN, DDoS, Intrusion Detection System, Random Forest, Machine Learning.

Proceedings Citation Format
Abdullahi, S.M., Mohammed, A., Ibrahim, R.Y. & Shamsuddeen, A. (2023): Detection of Algorithmically Generated Domain Names using Ensemble Machine Learning Technique. Proceedings of the Cyber Secure Nigeria Conference. Nigerian Army Resource Centre (NARC) Abuja, Nigeria. 11-12th July, 2023. Pp 27-34.
https://cybersecurenigeria.org/conference-proceedings/volume-2-2023/ dx.doi.org/10.22624/AIMS/CSEAN-SMART2023P4

1. INTRODUCTION

The world today relies so much on information technology in all facets. Cyber attackers leverage on this reliance to launch sophisticated cyber-attacks to compromise the integrity of data and information and to wreak havoc on victim computers. Using the DGA technique, cyber criminals generate a large number of malicious pseudo-random domain names within a short period of time. Thereafter, the attackers then use one of these domain names to resolve the Domain Name Service (DNS) address of the C&C server and to establish a secure communication with the attacker. Once this communication is established, the malware sends/receives data/instructions with the attacker.

Thereafter, the attacker seizes complete control of the compromised system and spreads malware (either a botnet or a ransomware). After the malware has been spread and the system or network hijacked, the botmaster uses the compromised system or network to target single or multiple computers within the network with the aim of either stealing confidential data or information, disabling or hijacking the system or network or using the breached system or network as a launchpad for further attacks. These attacks could be either distributed denial of service attacks, man in the middle attacks, phishing attacks, SQL injection attacks, etc. The research begins by extracting comprehensive set of features from the domain names. thereafter, the 4 algorithms were trained individually to make predictions. The models were trained and evaluated using a large dataset of domain names data. Results of the research shows the ensemble machine learning model having a high accuracy level with improved detection performance and reduced false positives.

2. RELATED WORKS

Wang et al. (2016) proposes a DGA botnet detection mechanism using the feature-based characteristics of social networks. In their proposed research, a filtering module, a clustering module, and a group identification module made up their suggested model. The filtering module separates known domains from unknown domains, the clustering module groups the hosts into a particular DGA algorithm and the group identification module identifies whether or not a candidate group belongs to a malicious domain or a normal domain. Abbink and Doerr (2017) investigated how well existing DGA detection algorithms performed when the domains produced by these DGAs were real dictionary words that are very similar to common or benign domain names. The outcome of their research shows that changing DGA names from randomly selected letters to dictionary words would have a considerable impact on the effectiveness of current DGA detection models.

Furthermore, Yang et al. (2018) proposed a random forest classifier for classifying wordlist-based DGAs that makes use of manually collected characteristics such as word frequency, part-of-speech tags, and word correlations. The outcome of this research shows that the random forest classifier was capable of accurately predicting and categorising the domains as either benign domains or harmful DGA domains. Wang and Guo (2021) describe a botnet based DGA which generates domain names by concatenating words randomly chosen from specific dictionaries to form malicious domains.

They presented a deep learning architecture to generate domain names that are difficult to distinguish from benign domain names. Their proposed method tried on some known classes of DGA malwares such as the Bamital, Banjori, and Suppobox.

Although several research works have been carried out on DGA detection and classification techniques as discussed above, most of them merely considers some common classes of botnets form their training and testing data sets. Also, most of the previous works done on DGA detection uses the Alexa top one million domain names data as their benign training dataset. Additionally, these research works further uses known features of these domain names in classifying them as either malicious domains or benign domains. This research however, combines both botnets and ransomwares DGA malwares and seeks to provide an ensemble machine learning approach for its detection and classification. The research further uses a different training dataset, i.e., the Cisco Umbrella top 1 million most visited domain names. furthermore, the research uses attributes that were extracted from the domain names data itself in trying to classify and detect whether a domain name is benign or malicious.

3. DATA COLLECTION AND FEATURE EXTRACTION

The training dataset for this research comprised 601,200 datasets, from which 200,000 are normal or benign domain names data, while 401,200 are maliciously generated DGA domain names data. The research uses the Cisco Umbrella top one million domain names data as test data for the benign or normal domain names. On the other hand, the malicious domain names training data was downloaded from DGArchive which is a collection of maliciously generated domain names by various classes of malware DGAs. It is offered by Fraunhofer FKIE and administered by Daniel Plohmann. The malicious training data comprised of 3 different classes of DGA botnets and 2 different classes of DGA ransomware families. The Table below presents the summary of the dataset used for this research.

Table 1: Sample Training Dataset

Malicious Sample Training Data		
S/No	DGA Family	Sample Data Size
1	Conficker	100,000
2	Bamital	100,000
3	Banjori	100,000
4	Cryptolocker	100,000
5	Dicrypt	1,200
Normal/Benign Sample Training Data		
1	Cisco Umbrella Top 1M	200,000
	Total	601,200

4. METHODOLOGY

The methodology adopted in this research works was similar to the Cross-Industry Standard Process for Data Mining (CRISP-DM) model as shown in Figure 1.

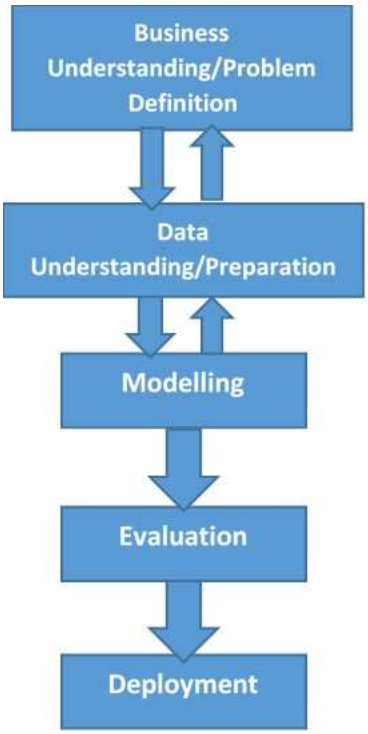

Figure 1: Cross-Industry Standard Process for Data Mining (CRISP-DM) model

Figure 1 shows the design overview through which this project was conducted. The problem was first investigated through a review of the existing literature to understand what have been done on DGA detection, how it was done, and the gaps existing. Thereafter, the test dataset was downloaded, pre-processed, and prepared before the model was trained and thereafter evaluated. The research was implemented using the R programming language. 10 features were extracted from the domain names data to help in classifying them as either benign or malicious domains. Some of the features extracted are the length of the domain name, whether or not it has numbers, special characters.

After the training data has been pre-processed for training, 4 different machine learning models were deployed for this research. The models were; naïve bayes, support vector machines, random forest and classification and regression tree model. Additionally, the models were trained twice; first using only 4 features and with a training dataset of 300,000 randomly selected out of the 601,200 available training datasets, and then, with 10 features and 300,000 randomly selected training datasets. The data was split into 80% training data and 20% for validation. The models were also trained using 10-fold cross validation for efficiency.

5. RESULTS DISCUSSION

Table 2 and 3 below shows the summary of the results obtained by all the models as well as the training time for the models.

Table 2: Summary of the results of the Models

Model		*Accuracy*	*Sensitivity*	*Specificity*	*Detection Rate*	*Balanced Accuracy*
Naïve Bayes	*10 Features*	*97.54%*	*97.48%*	*100%*	*94.92%*	*93.74%*
	4 features	*95.99%*	*90.15%*	*100%*	*96.93%*	*95.07%*
SVM	*10 Features*	*94.50%*	*98.36%*	*96.80%*	*92.28%*	*97.58%*
	4 features	*95.03%*	*98.89%*	*97.44%*	*92.78%*	*98.16%*
Random Forest	*10 Features*	*95.02%*	*99.05%*	*96.96%*	*92.69%*	*98.00%*
	4 features	*94.99%*	*98.71%*	*97.52%*	*92.43%*	*98.12%*
CART	*10 Features*	*96.64%*	*97.42%*	*95.76%*	*91.58%*	*96.59%*
	4 Features	*93.99%*	*98.71%*	*97.52%*	*93.43%*	*98.12%*

Table 3: Summary of the time taken for model training/model predictions

Model		*Training Time*	*Predictions Time*
Naïve Bayes	*10 Features*	*1 min*	*Instant within seconds*
	4 features	*1 min*	*Instant within seconds*
SVM	*10 Features*	*6 hrs*	*1 min*
	4 features	*4 hrs*	*1 min*
Random Forest	*10 Features*	*6 hrs*	*1 min*
	4 features	*5 hrs*	*1 min*
CART	*10 Features*	*2 mins*	*Less than 1 min*
	4 Features	*1 min*	*Less than 1 min*

5.1 Speed-Accuracy Trade Off

According to Zimmerman (2011), the speed-accuracy trade-off describes the complex relationship between a model's slow execution time and a model's ability to make fewer errors in its predictions, as compared to a model's fast execution time, and relatively making errors in its predictions. While both the speed of a model as well as its overall performance are important considerations in choosing a best fit model, there must be some kind of trade-offs between the speed and accuracy when comparing two or more models together as the output of the models may vary depending on many varying circumstances.

As regards to the accuracy of the models, there is no much of a difference in the results of the models when trained with both 4 and 10 features respectively albeit using same training dataset and same computing resources as shown in table above. However, there is a huge difference in the execution time of the models or models training time as shown in table 3 above.

6. CONCLUSSION AND FUTURE WORK

This research seeks to develop an ensemble machine learning technique for the classification and detection of algorithmically generated domain names. The research uses 4 different machine learning models. The models were trained using 4 and 10 features respectively with 300,000 randomly selected training datasets. While all the models performed excellently well in terms of accuracy with all the models having an accuracy level of more than 90%, some models performed better especially as regards the execution time. Also, no malicious domain name was wrongly classified as benign by the Naïve Bayes model when trained with both 4 and 10 features respectively. This is, however, not the case with SVM and Random Forest models where some malicious domain names were wrongly classified as benign domain names. From a security perspective, it is better for a model to wrongly classify a benign domain name as a malicious domain than for a malicious domain name to be wrongly classified as benign domain, as this could result to serious security breaches with severe consequences. Hence, the Naïve Bayes model is hereby considered the best fit model in this research both in terms of its speed of execution and accuracy level, and thus recommended for deployment. The future work for this research work could be implemented using live domain names data as training datasets.

REFERENCES

1. Abbink, J., & Doerr, C. (2017). Popularity-based detection of domain generation algorithms. *Proceedings of the 12th International Conference on Availability, Reliability and Security*.
2. Akarsh, S., Sriram, S., Poornachandran, P., Menon, V. K., & Soman, K. P. (2019). Deep learning framework for domain generation algorithms prediction using long short-term memory. *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*.
3. Arntz, P. (2016, December 6). *Explained: Domain generating algorithm*. Malwarebytes. https://blog.malwarebytes.com/security-world/2016/12/explained-domain-generating-algorithm/
4. Asher-Dotan, L. (n.d.). *What is domain generation algorithm: 8 real world DGA variants*. Cybereason.com. Retrieved June 5, 2023, from https://www.cybereason.com/blog/what-are-domain-generation-algorithms-dga
5. Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Annals of Statistics*, *47*(2), 1148–1178. https://doi.org/10.1214/18-aos1709
6. Chen, Y., Yan, S., Pang, T., & Chen, R. (2018). Detection of DGA domains based on support vector machine. *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*.

7. Communication, B., & Ollmann, G. (n.d.). *Targeted protection against targeted attacks*. Technicalinfo.net. Retrieved June 5, 2023, from http://www.technicalinfo.net/papers/PDF/WP_Botnet_Communications_Primer_(2009-06-04).pdf
8. Detection of algorithmically generated domain names used by botnets: A dual arms race. (n.d.). In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*.
9. *Deutsch - Fraunhofer FKIE*. (2023, May 30). Fraunhofer.de. https://www.fkie.fraunhofer.de
10. Gupta, S. (2020, February 28). *Pros and cons of various Machine Learning algorithms*. Towards Data Science. https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6
11. Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: A brief primer. *Behavior Therapy*, *51*(5), 675–687. https://doi.org/10.1016/j.beth.2020.05.002
12. Katsimpokis, D., Hawkins, G. E., & van Maanen, L. (2020). Not all speed-accuracy trade-off manipulations have the same psychological effect. *Computational Brain & Behavior*, *3*(3), 252–268. https://doi.org/10.1007/s42113-020-00074-y
13. Kumar, A. D., Thodupunoori, H., Vinayakumar, R., Soman, K. P., Poornachandran, P., Alazab, M., & Venkatraman, S. (2019). Enhanced domain generating algorithm detection based on deep neural networks. In *Deep Learning Applications for Cyber Security* (pp. 151–173). Springer International Publishing.
14. Lalmuanawma, S., Hussain, J., & Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons, and Fractals*, *139*(110059), 110059. https://doi.org/10.1016/j.chaos.2020.110059
15. Li, Y., Xiong, K., Chin, T., & Hu, C. (2019). A machine learning framework for domain generation algorithm-based malware detection. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 32765–32782. https://doi.org/10.1109/access.2019.2891588
16. Liu, Q., Zhang, J., Liu, J., & Yang, Z. (2022). Feature extraction and classification algorithm, which one is more essential? An experimental study on a specific task of vibration signal diagnosis. *International Journal of Machine Learning and Cybernetics*, *13*(6), 1685–1696. https://doi.org/10.1007/s13042-021-01477-4
17. Mabon, W. (2020, February 27). *Cybersecurity news: Get the latest trends & threats*. Cisco Umbrella. https://umbrella.cisco.com/blog/cisco-umbrella-1-million
18. Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M. J., & Flach, P. (2021). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, *33*(8), 3048–3061. https://doi.org/10.1109/tkde.2019.2962680
19. Namgung, J., Son, S., & Moon, Y.-S. (2021). Efficient deep learning models for DGA domain detection. *Security and Communication Networks*, *2021*, 1–15. https://doi.org/10.1155/2021/8887881
20. Palaniappan, G., Sangeetha, Rajendran, B., Sanjay, Goyal, S., & Bindhumadhava. (2020). Malicious domain detection using machine learning on domain name features, host-based features and web-based features. *Procedia Computer Science*, *171*, 654–661. https://doi.org/10.1016/j.procs.2020.04.071

21. Ray, S. (2017, September 11). *Naive Bayes classifier explained: Applications and practice problems of Naive Bayes classifier*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained
22. Ren, F., Jiang, Z., Wang, X., & Liu, J. (2020). A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network. *Cybersecurity*, *3*(1). https://doi.org/10.1186/s42400-020-00046-6
23. Sari Oktapia Ningse, W. R., Sumarno, S., & Nasution, Z. M. (2022). C4.5 algorithm classification for determining Smart Indonesia Program Recipients at MIS Al-Khoirot. *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, *1*(1), 65–76. https://doi.org/10.55123/jomlai.v1i1.165
24. Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, *181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199
25. Sood, A. K., & Zeadally, S. (2016). A taxonomy of domain-generation algorithms. *IEEE Security & Privacy*, *14*(4), 46–53. https://doi.org/10.1109/msp.2016.76
26. Sridharan, M. A. (2018, September 25). *CRISP-DM - A framework for data mining and analysis*. Think Insights. https://thinkinsights.net/data-literacy/crisp-dm/
27. Wang, Z., & Guo, Y. (2021). Neural networks-based domain name generation. *Journal of Information Security and Applications*, *61*(102948), 102948. https://doi.org/10.1016/j.jisa.2021.102948
28. Yu, B., Pan, J., Gray, D., Hu, J., Choudhary, C., Nascimento, A. C. A., & De Cock, M. (2019). Weakly supervised deep learning for the detection of domain generation algorithms. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 51542–51556. https://doi.org/10.1109/access.2019.2911522