Semi-Supervised Detection of Algorithmically Generated Domains using Neural Network-Based Autoencoders

Federico Cruciani[a], Samuel Moore[a], Bronagh Quigley[a], Chris Nugent[a] and Sadiq Sani[b]

[a] *School of Computing, Ulster University, Shore Road, Newtownabbey, BT37 0QB, Northern Ireland*
[b] *BT Research, Adastral Park, Martlesham, IP5 3RE, Ipswich, UK*

Abstract

Domain Generation Algorithms (DGA) are typically used by recent botnets to communicate with their command-and-control server, thus exacerbating the complexity of detecting them compared to older botnets using static IP addresses. As such, recent studies have been experimenting with different approaches to detect algorithmically generated domains using a variety of methods, including Deep Learning. This paper presents a Deep Learning approach based on autoencoders as a semi-supervised method requiring only legitimate domains for training. Semi-supervised methods have an advantage over supervised methods in that they require no labelled DGA data. The proposed autoencoder structure is based on a Neural Network (NN) processing the frequency of 2-grams in domain names. The method has been compared with supervised machine learning methods and cross-validated on a second unseen dataset to evaluate the generalization of results. Results confirmed an F-score of 73% on DGA detection outperforming a NN based on letter frequencies and a Random Forest approach based on n-grams scoring 71% and 65% respectively.

Keywords

Domain Generation Algorithms, Autoencoder, Semi-supervised learning, Cybersecurity

1. Introduction

In our increasingly connected world, cyber-security has become a fundamental field. Moreover, the rapid acceleration and growth of the Internet of Things (IoT) has led to a dramatically increased attack surface, making our networks more vulnerable to malicious attacks. The consequences of undetected attacks can be severe, now that our computer networks are responsible for hosting significant amounts of mission critical infrastructure. These consequences can range from partial reduction of service, all the way to complete system failure. As such, there is a critical need to develop systems and techniques to detect network intrusions before negative consequences occur. One such malicious attack, particularly pervasive in IoT environments due to its host of constrained devices, is a Botnet attack. A Botnet attack involves a machine

AI-CyberSec 2021: Workshop on Artificial Intelligence and Cyber Security, December 14, 2021, Cambridge, UK.

f.cruciani@ulster.ac.uk (F. Cruciani); s.moore2@ulster.ac.uk (S. Moore); b.quigley@ulster.ac.uk (B. Quigley); cd.nugent@ulster.ac.uk (C. Nugent); sadiq.sani@bt.com (S. Sani)
0000-0002-1870-0203 (F. Cruciani); 0000-0003-3205-3310 (S. Moore); 0000-0002-6161-3754 (B. Quigley); 0000-0003-0882-7902 (C. Nugent); 0000-0001-9784-8398 (S. Sani)

CEUR Workshop Proceedings (CEUR-WS.org)

infected with malware seeking to establish communication with a Command and Control (C&C) server. This C&C machine will then deliver instructions to the infected host in order to carry out actions such as retrieving unauthorized information from the victim's target network. In more extreme circumstances, Botnet attacks can even be used to infect large numbers of victim machines in order to instigate Distributed Denial of Service (DDoS) attacks. These DDoS attacks can bring down significant pieces of infrastructure, such as was the case when the Mirai Botnet attack brought down the internet connectivity for most of the East coast of the United States in 2016 [1]. Often, these botnet attacks use a Domain Generation Algorithm (DGA) to establish the communication between the victim machine and the C&C [2, 3]. Therefore, early detection of DGA activity in computer networks is a vital step in ensuring that our networks are safe from such attacks. A DGA is capable of generating a list of domains, using an algorithm which often uses a public seed, such as the weather forecast or the current date. This list of domains can then be registered by an attacker, allowing the infected machines to successfully resolve the Domain Name System (DNS) query on the target network. Through the resolution of this domain, the C&C machine is able to establish communication with the victim host, which is inside the target network. As such, preventing this domain registration from taking place, is a crucial area of interest for cyber security engineers. Naturally, the consequences of missing the detection of an Algorithmically Generated Domain (AGD) in a large network could be quite severe. Therefore, for any designed classification mechanism having a high recall on the AGDs is of great importance.

This research presents a Deep Learning (DL) approach to detect AGDs using Autoencoders (AEs). The contributions of this research are as follows; firstly, we propose a semi-supervised approach based on AEs for the design of a DGA detection system. Secondly, we propose the use of a thresholding method on the reconstruction error of the AE in order to trigger alerts while providing information on the confidence interval of the detection. Finally, our proposed model achieves a higher recall statistic than the baseline approach. The remainder of this paper is structured as follows. Section 2 provides a concise overview of related work. Section 3 provides details on the proposed method. The evaluation methodology is described in Section 4. Results and discussion are reported in sections 5 and 6, respectively. Finally, Section 7 draws the conclusion of the experiment while introducing future work.

2. Related Work

In recent years, an increasing number of studies have been undertaken to detect domain names produced by DGAs via different approaches. Authors in [4] proposed an algorithm resilient to feature change that used 16 bit representation of domain names and used an AE to classify the domain using a Neural Network (NN) and Support Vector Machine (SVM). A work from [5] focused on the detection and generation of DGA leveraging the concept of Generative Adversarial Networks (GANs). They used GANs to build a DL based DGA designed to intentionally bypass a DL based detector. Their results show that domains generated from GANs to bypass the GANs detector, also bypassed a Random Forest (RF) classifier. Another approach used to detect AGDs is through the use of n-grams, *i.e.* contiguous sequences of a n symbols from a given text. In the case of detecting AGDs, n-grams are typically used to compare their frequency of occurrence with their normal distribution in known domains. Wang *et al.* in [6] proposed a method to detect the use of AGDs based on visualization and n-grams methods. Xu *et al.* [7] proposed a novel combined n-grams and Convolutional Neural Network (CNN) approach to DGA detection, n-gram Combined Character Based Domain Classification (n-CBDC). Results show an accuracy of 98.69% for detection of AGDs. Authors in [8] used a supervised Machine Learning (ML) approach to detect DGA domain names. Their approach focuses on features which state the similarity between the 2-grams and 3-grams in a single unclassified domain name. Results show a good accuracy was achieved which outperformed some of the state-of-the-art featureless classification methods based on DL. In [9], a RF approach and a combination of statistical features and n-gram based features were used for AGD for detection. Authors in [10] proposed a ML model to detect the output of DGA. measuring the randomness in the characters of a domain name from the DNS request packet only. Results show their approach can effectively detect AGDs domain names by several malware types with an accuracy of 98%. Authors in [11] propose a ML approach using clusters, to detect AGDs. A two-step approach is introduced, where the first step queries the automatically generated domains to detect a bot looking for the C&C Centre. The second step involves analysis of DNS requests resolved in the same time interval. This approach was evaluated in an experiment with an ad-hoc network with injected DGA, and later a Local Area Network (LAN) of a company. This approach successfully detected AGDs within the first experiment, and discovered an infected host within the network in the second experiment.

DL is a popular approach used to detect AGDs with a number of DL approaches described in the literature. A work by [12] proposed a DL framework using Long Short-Term Memory (LSTM) to detect domain names generated using DGA. Binary classification was performed on benign and DGA names using two datasets, where the LSTM model results in 98.7% and 71.3%. The multiclass classification was performed using 20 DGA names and resulted in an accuracy of 68.3% and 67% on the same datasets. Authors in [13] proposed two novel models based on CNN to detect real-time DGA use. They compared results against several Recurrent Neural Network (RNN) and CNN. Their comparison shows one of the proposed models performed as well as other established DL models in the literature. Amara *et al.* [14] propose a CNN based architecture using a character level domain name encoding approach for detecting DGA generated domain names. Results show the approach successfully detected DGA generated domain names with an accuracy of 99.7% and 97.1% across two datasets and performed well when compared against other state of the art DL approaches. Based on the literature, lab based experiments that detect AGD achieve better results than that of real world experiments. Within real world DGA detection, the challenge of model generalization with real-world data still remains. As such, our experiments aimed at evaluating the generalization abilities of the proposed model with unseen data.

3. Implementation

The proposed solution is based on a deep AE [15]. As mentioned in Section 2, AEs are a particular type of NN having the output layer of the same size of the input layer and two symmetric sets of hidden layers: (i) an encoder connected to the input, followed by (ii) a decoder part preceding the output layer. Originally introduced as a DL approach for dimensionality reduction [15], the use of AEs has rapidly found an increasing number of applications, including machine translation [16], and anomaly detection [17]. In our implementation, a NN AE was used in a similar fashion of the anomaly detection application. In such a case, the AE is trained to try and reconstruct a replica of the input data in the output layer, *i.e.* the AE is trained to implement a mapping function $f : X \rightarrow X$ generating a replica $\hat{x}$ of x with $x, \hat{x} \in X$:

$$\hat{x} = f(x) = decoder(encoder(x))$$

where X is the domain in which a domain name is represented. The training process of an AE therefore can be seen as an optimization problem having the goal of minimizing the difference between the input and its reconstructed replica [18] as in Equation 1.

$$\min_f ||x - f(x)|| \tag{1}$$

As in the anomaly detection case, an AE trained on legitimate domain names can be expected to produce a higher reconstruction error $x - \hat{x}$ for domains presenting different characteristics from the ones used in the training data. The Domain Alphabet DA, defined as the set of admissible characters in a domain name, includes all alphabet letters, numerical digits and a limited number of extra characters:

$$DA = \{a, b, c, d, e, \ldots y, z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., \_, -\}$$

where letters are case insensitive ($a == A$). The set of symbols composing the DA were used to define to different representation of domain names based on letter frequency and n-grams.

3.1. Letter frequency

Using the cardinality of the set of legitimate characters $|DA| = 39$, one of the simplest representations of a domain name is its representation in terms of the frequency of appearance of each symbol. In our implementation each domain name can be represented as a vector $x \in \mathbf{R}^N$, with N being $N = |DA|$

$$x = \frac{[x_1 \quad x_2 \quad \ldots \quad x_N]}{|D|}$$

where x_i is the number of occurrences of the $i-th$ character in DA normalized using the total number of characters $|D|$ in a domain name D. In this simple representation, each domain name is mapped into a vector of 39 scalar values in $[0, 1)$. This approach, however, by only taking into account the number of occurrences of each character, can only learn the distribution of frequency of characters and does not maintain any notion related to the sequence of letters by their order of appearance. Such a feature space is therefore invariant to all the possible string permutations obtainable using the same set of characters, regardless of their order. This limitation, as briefly mentioned in Section 2, has been addressed by using different approaches: one attempting to learn legitimate sequences of characters as in the case of LSTM, or by using n-grams. In our implementation, the use of AEs applied to the frequency of n-grams was considered.

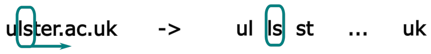

Figure 1: Example of reconstruction of 2-grams sequence in a domain name.

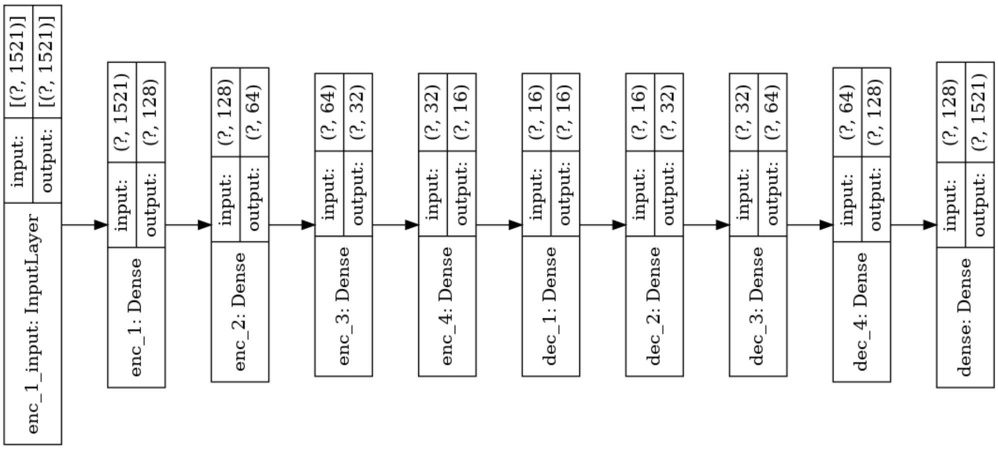

Figure 2: MLP AE structure.

3.2. Implementation based on 2-grams

Trying to cope with the limitations of the letter frequency method an n-grams implementation has also been evaluated. In our experiment, the case of using 2-grams was considered, thus building an input domain in $X = \mathbf{R}^{N^2}$. As illustrated in Fig. 1, the occurrence of 2-grams in the domain name is obtained with a sliding window approach reading the string from left to right with 50% overlap. As in the generic case of n-grams, even if not aiming at learning sequences of legitimate characters, the method preserves part of the order information in the sequence of obtained 2-grams, thus creating a different representation for strings containing the same set of symbols in different orders.

Other approaches, such as [9], exclude the top level domain and any subdomains starting from the third level domain. In our implementation, for the pre-processing of domain names into vectors $x \in \mathbf{R}^{N^2}$ the top three domain levels were considered.

3.3. The Autoencoder model

The AE model was implemented using Keras [19]. The model consists of an encoder and a decoder block having 4 hidden layers each and an output layer having the same size as the input layer $NxN = 1521$.

The model was trained on the Majestic Million dataset with the goal of training the network to reconstruct most common characteristics on the 2-grams feature space for the top 1M legitimate domains. The training process was performed for 100 epochs using 20% domains as validation and with an update criterion saving models performing minimum loss on the validation set.

Table 1
Evaluation Dataset

Class	Bot	Domain
DGA	gozi	mortiscontrastatim.com
DGA	corebot	cvyh1po636avyrsxebwbkn7.ddns.net
Legit	Alexa	plasticbags.sa.com
Legit	Alexa	mzltrack.com
...	...	...

The threshold value for detecting a suspected AGD was set as $\mu + \sigma$ with μ and σ being the mean and the standard deviation of the loss on the training set calculated as $\mathbf{L} = ||x - \hat{x}||$.

4. Evaluation Methodology

The experiment explored the use of an Multilayer perceptron (MLP) AE for DGA detection, comparing the proposed approach with a RF approach [9], and the same AE structure using the simplified input feature space built using letter frequency. In [9], the length of the domain being examined and its entropy were used as statistical features together with two sets of n-grams features, the first being based on similarity between the distribution of n-grams in the Alexa top 1 million websites, and the second based on occurrence of n-grams in a list of common words. All methods were compared using an *unseen* dataset [8] including legit domains and AGDs generated by 25 DGA families. An illustratory fragment of the evaluation dataset [20] is reported in Table 1.

The evaluation dataset contains more than 600.000 domains, with approximately 50% of those being legitimate domains and the other 50% consisting in DGA generated by 25 bots for approximately 13.000 samples each. Cross-validation on the same unseen dataset allows the generalization abilities of the models to be assessed in working-like conditions, simulating their use on real-world data.

5. Results

Table 2 reports results obtained using the RF approach [9], exhibiting a high recall for legitimate domains and high precision for DGA 93%, for which, however, the recall was observed to be 50%. Table 3 reports results obtained using the proposed AE on the simple representation based on letter frequency. This model showed significantly higher recall on DGA (80%), however, at the expense of lower recall (56%) on legitimate domains. The use of 2-grams in the proposed model has been observed to further improve results using the AE approach, with the highest recall on DGA obtained with the highest precision on legitimate domains (75%), as reported in Table 4.

Table 2
Classification Report 3-grams RF

	Precision	Recall	F1-score	Support
Legit	0.66	**0.96**	0.78	337398
DGA	0.93	**0.50**	0.65	337500
Macro-average	0.80	0.73	0.72	674898

Table 3
Classification Report letter frequency MLP-AE

	Precision	Recall	F1-score	Support
Legit	0.74	**0.56**	0.64	337398
DGA	0.65	**0.80**	0.71	337500
Macro-average	0.69	0.68	0.68	674898

Table 4
Classification Report 2-grams MLP-AE

	Precision	Recall	F1-score	Support
Legit	**0.75**	0.61	0.68	337398
DGA	0.67	**0.80**	0.73	337500
Macro-average	0.71	0.71	0.70	674898

6. Discussion

The results obtained confirmed the detection of AGDs is a challenging task due to the great variety of characteristics of an increasing number of bot families. Among them, surely, dictionary based DGAs, whose output inherently presents similar characteristics to the legitimate ones, pose the greatest obstacle. This overlap in the feature space between certain DGAs and legitimate ones affects generalization abilities of models when dealing with real-world data. This is particularly relevant when looking at the comparison of recall values of the legitimate and the DGA class. In particular, the RF based model [9], despite having a high precision in detecting DGA (96%) produced a 50% recall, in contrast with a 96% recall on the legitimate class at the expense of a significantly lower precision on the legitimate class. The approach in [9], therefore, was found to detect DGA with high precision but with most of dictionary-DGA being mislabeled as legitimate. The two AE-based approaches, on the other hand, outperformed the RF based model in detecting DGA, but at the expense of a higher number of false positives. The process of model selection, in this regard, depends on the goal of the final application that can change the perspective on the interpretation of obtained results. For our purposes, we prioritized the detection of a possible situation of risk over the generation of a false positive. As mentioned in Section 3.3, the proposed threshold was defined considering mean and standard deviation of the reconstruction error on the training dataset. Such an approach, by definition, will generate false positive detection of a DGA for legitimate domains in the tails of the distribution. Such an approach also allows to define a scoring mechanism evaluating an increasing risk of based on confidence intervals, e.g. 95.45% when $\mathbf{L} \leq \mu + 2\sigma$ and 99.73% with $\mathbf{L} > \mu + 3\sigma$.

7. Conclusion

This paper reported on an experiment evaluating the generalization abilities of AEs for DGA detection based on n-grams. The proposed method was cross-evaluated under realistic conditions and compared with a RF-based model and an AE with the same characteristics of the proposed model based on letter frequency. On the one hand, the experiment provided encouraging results highlighting the potentials of the proposed method. On the other hand, however, one of the limitations of the evaluation was that due to time-constraints a proper optimization of the AE model was not performed. Future work will involve further experiments in order to optimize the topology of the NN structure (number of layers and neurons), as well as other hyperparameters such as the number of epochs for training, the learning rate (which was left at default 0.01 for the experiment). Despite the lack of optimization in this regard, the proposed method was proven to generalize well on unseen data while preserving a trade-off between detection of AGDs and the precision in detecting legitimate domains. The experiment demonstrated that semi-supervised approaches can challenge supervised approaches and without the need of any labeled AGDs for training. This is a characteristic of major relevance considering the fact that new DGAs are always emerging, and fully supervised approach may struggle to detect new DGA types for which they have not been trained. Thus, an optimized semi-supervised approach has the potential of providing better generalization results when dealing with real-world unseen AGDs compared to a supervised approach relying on detecting learned features of a set of DGA families. Finally, the proposed method can be used not only to detect AGDs, however, also to provide a metric for assessing the confidence interval of detection with respect with the normal distribution of legitimate domains.

Acknowledgments

This research is supported by the BTIIC (British Telecom Ireland Innovation Centre) project, funded by BT and Invest Northern Ireland.

References

[1] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, et al., Understanding the mirai botnet, in: 26th {USENIX} security symposium ({USENIX} Security 17), 2017, pp. 1093–1110.

[2] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, E. Gerhards-Padilla, A comprehensive measurement study of domain generating malware, in: 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 263–278.

[3] R. Sivaguru, C. Choudhary, B. Yu, V. Tymchenko, A. Nascimento, M. De Cock, An evaluation of dga classifiers, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 5058–5067.

[4] B. Dahal, Y. Kim, Autoencoded domains with mean activation for dga botnet detection, in: 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3), 2019, pp. 208–212. doi:10.1109/ICGS3.2019.8688037.

[5] H. Anderson, J. Woodbridge, B. Filar, Deepdga: Adversarially-tuned domain generation and detection, Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (2016).

[6] J. Selvi, R. Rodríguez, E. Olivas, Detection of algorithmically generated malicious domain names using masked n-grams, Expert Systems with Applications 124 (2019). doi:10.1016/j.eswa.2019.01.050.

[7] C. Xu, J. Shen, X. Du, Detection method of domain names generated by dgas based on semantic representation and deep neural network, Computers and Security 85 (2019) 77–88. doi:10.1016/j.cose.2019.04.015.

[8] A. Cucchiarelli, C. Morbidoni, L. Spalazzi, M. Baldi, Algorithmically generated malicious domain names detection based on n-grams features, Expert Systems with Applications 170 (2021) 114551.

[9] A. Taylor, Flare: An analytical framework for network traffic and behavioral analytics, 2017. URL: https://github.com/austin-taylor/flare.

[10] A. O. Almashhadani, M. Kaiiali, D. Carlin, S. Sezer, Maldomdetector: A system for detecting algorithmically generated domain names with machine learning, Computers and Security 93 (2020). doi:10.1016/j.cose.2020.101787.

[11] F. Bisio, S. Saeli, P. Lombardo, D. Bernardi, A. Perotti, D. Massa, Real-time behavioral dga detection through machine learning, in: 2017 International Carnahan Conference on Security Technology (ICCST), 2017, pp. 1–6. doi:10.1109/CCST.2017.8167790.

[12] S. Akarsh, S. Sriram, P. Poornachandran, V. K. Menon, K. P. Soman, Deep learning framework for domain generation algorithms prediction using long short-term memory, in: 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), 2019, pp. 666–671. doi:10.1109/ICACCS.2019.8728544.

[13] D. S. Berman, Dga capsnet: 1d application of capsule networks to dga detection, Information (Switzerland) 10 (2019). doi:10.3390/info10050157.

[14] D. Amara, H. Thodupunoori, V. Ravi, S. Kp, P. Poornachandran, M. Alazab, S. Venkatraman, Enhanced Domain Generating Algorithm Detection Based on Deep Neural Networks, 2019, pp. 151–173. doi:10.1007/978-3-030-13057-2_7.

[15] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, science 313 (2006) 504–507.

[16] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259 (2014).

[17] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis, 2014, pp. 4–11.

[18] C. Zhou, R. C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 665–674.

[19] F. Chollet, et al., Keras, https://keras.io, 2015.

[20] C. Morbidoni, Dga domains, 2020. URL: https://github.com/chrmor/DGA_domains_dataset.