AI-based Detection of DNS Misuse for Network Security

Irina Chiscop
irina.chiscop@tno.nl
Department of Cyber Security Technologies
Netherlands Organisation for Applied Scientific Research
The Hague, The Netherlands

Francesca Soro
francesca.soro@ait.at.ac
Center for Digital Safety and Security
AIT Austrian Institute of Technology
Vienna, Austria

Paul Smith
paul.smith@ait.at.ac
Center for Digital Safety and Security
AIT Austrian Institute of Technology
Vienna, Austria

ABSTRACT

Threat hunting and malware prediction are critical activities to ensure network and system security. These tasks are difficult due to increasing numbers of sophisticated malware families. Automatically detecting anomalous Domain Name System (DNS) queries in operational traffic facilitates the detection of new malware infections, significantly contributing to the work of security practitioners. In this paper, we present two AI-based Domain Generation Algorithm (DGA) detection and classification techniques – a *feature-based* one, leveraging classic Machine Learning algorithms and a *featureless* one, based on Deep Learning – specifically intended to aid in this task. Both techniques are designed to be integrated in operational environments, dealing with hundreds of thousands to millions of new malware samples per day. We report the implementation details, the classification performance, the advantages and shortcomings for both techniques, as well as experiences from the deployment of this system in an industrial environment. We show that both techniques reach more than the 90% of accuracy in the case of binary DGA detection, with a slight degradation in performance in the multi-class classification case, in which the results strongly depend on the malware type.

CCS CONCEPTS

• **Networks** → **Network monitoring**; **Security protocols**; • **Information systems** → **Data analytics**.

KEYWORDS

Intrusion Detection, Malware Identification, Threat Intelligence

ACM Reference Format:
Irina Chiscop, Francesca Soro, and Paul Smith. 2022. AI-based Detection of DNS Misuse for Network Security. In *Native Network Intelligence (NativeNI '22), December 9, 2022, Roma, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3565009.3569523

NativeNI '22, December 9, 2022, Roma, Italy

ACM ISBN 978-1-4503-9887-9/22/12…$15.00
https://doi.org/10.1145/3565009.3569523

1 INTRODUCTION

Domain Generation Algorithms (DGAs) are used by malware authors to automatically generate very large numbers of domain names. Knowledge of a DGA, a time period, and the input seed to the DGA allows a cyber-criminal to determine what unique domain name(s) will be required to be registered on a particular day, in order to maintain Command and Control (C2) communications with malware-infected computers, called *bots*. On the defenders' side, it is very difficult to predict what C2 domains will be used by malware, until live malware traffic can be observed in an isolated environment. However, at this point in time, the cyber-criminal will likely already have registered the C2 domain for that day and ensured they have gained or maintained control of their network of bots (*botnet*). Only one of the many DGA-generated domains needs to be registered, since a bot will try and access all the domains generated for the specific time interval, until it resolves the IP address of the C2 server.

Well-known malware instances that use DGAs include the Gameover Zeus banking trojan [10], which generated 1000 possible C2 domains per *week*, and the CryptoLocker ransomware [8] that generated 1000 possible C2 domains per *day*. Despite the progress on malware analysis techniques and machine learning-based detection methods, maintaining connectivity to compromised computers using DGAs remains a popular technique for threat actors. The more recent FluBot Android malware [9] and the 2020 SolarWinds Sunburst [7] attack campaign, which affected multiple US government institutions, was only discovered *eight* months after its initial breach, show that there exists a gap between current detection systems and the malware trends in practice.

To address this issue, we have developed two AI-based DGA detectors, a feature-based and a featureless approach to *(i)* discriminate between benign and DGA domains; and *(ii)* perform a finer-grained classification, recognizing which malware family generated each domain. The featureless approach involves a novel use of Temporal Convolutional Networks (TCNs) [2] to this problem domain. An evaluation of the two approaches indicates that overall detection performance is good, with accuracy scores for binary classification reaching more than 90%. However, we have found that for multi-class classification, detection performance differs between the approaches, depending on the DGA. This finding points to the need for combining these approaches to get better overall performance. A prototype of the featureless DGA detector has been deployed in an operational environment, identifying and classifying domains at scale (a few million domain names daily). This

deployment has resulted in the identification of new DGAs, thus helping to bridge the gap between detection capabilities in the field and changing threat actor practices.

This paper is organized, as follows: Section 2 provides an overview of studies in the DGA detection literature. The proposed feature-based and featureless approaches are detailed in Section 3. The results both in a test and operational environment are discussed in Section 4. Section 5 provides some conclusions and outlines further research.

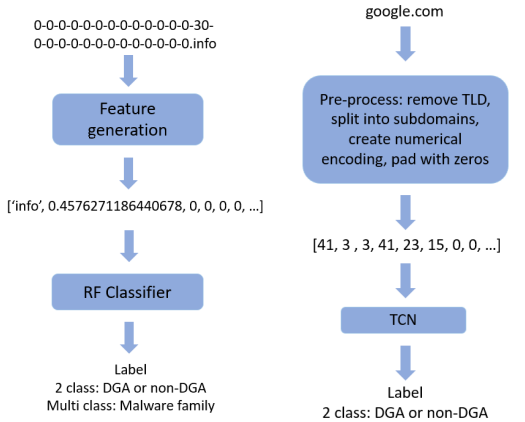
2 RELATED WORK

The existing literature on DGA detection focuses primarily on machine learning methods that can either classify a given domain as benign or malicious, or determine the malware family to which the domain belongs. According to one of the most recent literature surveys on this topic by Sivaguru *et al.* [20], DGA detection approaches can be split into two categories, depending on the input data: feature-based or featureless approaches.

Feature-based approaches rely on a variety of features being fed to supervised or unsupervised algorithms. Vranken and Alizadeh [25] have surveyed some of the most commonly used features in the DGA literature. They identified 97 features and grouped them into five categories: domain-, subdomain-, or character-related, linguistics and n-grams analysis (e.g. ratio of consecutive vowels, pronounceable n-grams, length of the top-level domain, etc.). Several contributions [1, 12, 18, 19] use well-established supervised algorithms like a Support Vector Machine (SVM), Naïve Bayes and Random Forest (RF) on similar datasets. Selvi *et al.* [19] improve the classification process by means of an additional character preprocessing step, to achieve better accuracy. A more recent study by Cucchiarelli *et al.* [3] proposes using Kullback–Leibler divergence and the Jaccard index of 2- and 3-grams as input features for three supervised classifiers. Some contributions [14, 21, 22] further enlarge their datasets with "side-information" from DNS records and network traffic. Other works, like that from Li *et al.* [14] follow a two-step approach: after the detection phase, DBSCAN [5] is applied on the domains recognized as malicious, to cluster those generated by the same DGA.

Recently, the number of studies addressing featureless DGA detection approaches have increased significantly. In featureless approaches, a character encoding is applied to the domain name, and the resulting numerical sequence is used as input for the respective algorithm. Such procedure allows to avoid the feature extraction step - generally a very costly process, especially when dealing with large datasets. In general, supervised (deep) learning methods are the most popular both for binary classification (benign vs. DGA), as well as multi-class (DGA malware family) classification. Amongst the neural networks proposed for string classification tasks[1] there are architectures containing a single Long Short-Term Memory (LSTM) layer [26], multiple parallel Convolutional Neural Network (CNN) layers [17], a CNN-LSTM encoder-decoder structure [24], stacked CNN layers [27, 28], and stacked residual blocks (ResNet) [4]. Drichel *et al.* [4] perform a comparison between some of the mentioned neural networks, for both binary and malware family classifications. They argue that ResNet models outperform other architectures in terms of accuracy and the required training time.

[1]Besides domain names, we also consider URL, social media text etc. classification since the machine learning goal is essentially the same.



(a) Feature-based approach. **(b) Featureless approach.**

Figure 1: Domain classification workflow.

Some papers combine the feature-based and the featureless approach. The pipeline designed by Liang *et al.* [15] chooses one over the other depending on the length of the domain: if the domain is short (3-6 characters), embeddings are extracted and fed to a side-by-side multi-way convolution; moderate length domains (7-20 characters) are embedded to a bi-dimensional Right Shifted Tensor (RTS) and fed to a CNN, while for longer domains a feature-based approach based on linguistic features and Random Forest is chosen.

3 DGA DETECTION APPROACHES

In this section, we describe the implementation details for each one of the developed approaches. Figure 1 depicts both workflows. The same procedure is used both for the binary and multi-class classification tasks.

3.1 Feature-based approach

As discussed in Section 2, the feature-based approach exploits a set of features that describe the main characteristics of each domain name. Overall, we extract eleven features to feed to a RF classifier: the Domain Length and the TLD, as often domains of the same length and TLD are generated within the same DGA family; the ratio of numerical and special characters (e.g. &,+,*,-, etc.), usually higher in DGA domains; the ratio of meaningful (i.e. existing in the English vocabulary) 3-, 4- and 5-grams, usually higher in benign domains, as they are not randomly generated; and finally the string distance (Levenshtein) statistics, i.e. Mean, Standard Deviation, Maximum and Median of sample distance with respect to the rest of the dataset. The idea behind this feature set is that samples belonging to families with domains from the same root are closer to each other than to the rest of the dataset. Note that the feature generation phase is significantly slowed down by the extraction of the pairwise Levenshtein distance. The classic Levenshtein distance has a complexity of $O(n^2)$, given strings of length n, which causes

a drastic increase in the feature computation time as the dataset enlarges.

3.2 Featureless approach

The featureless approach used in this work relies on a TCN, which receive as input character encodings of the domain names. TCNs are a category of convolutional networks that are particularly suitable for modelling long-term dependencies in sequential data [2][23]. They have been shown to outperform recurrent architectures on many different sequence classification tasks, such as the adding problem and image classification on sequential MNIST and P-MNIST [2]. The classification process is depicted in Figure 1b.

First, pre-processing steps are applied to the domain name. The TLD is removed, the subdomains are extracted and then each subdomain receives a numerical encoding. In other words, each character is mapped to a positive integer. The list of characters that are considered includes all digits, letters in the English alphabet (no distinction is made between upper and lower case) and symbols ('-','.'). Finally, this numerical encoding is padded with zeros until a maximum specified domain length is reached. This last step ensures that all input data samples have the same size. This encoding is then passed to the TCN, which can classify a domain as generated by a DGA or not (for binary classification), or classify it according to the malware that created it (for multi-class classification).

The TCN block in the workflows shown in Figure 1b corresponds to a neural network consisting of the following items: an embedding layer that projects the input sequences into a higher dimension, a single TCN residual block, and an output layer that produces the result for the given input (the predicted class based on a sigmoid function). The model was trained for 50 epochs using a small portion of the data as a validation set. A stopping criterion was imposed to cease training if the validation loss did not improve in the previous three epochs.

4 EVALUATION RESULTS

In this section, we discuss the results obtained by both approaches in our testing environment. We conclude this section with a discussion and a description of our contribution in operational deployments.

4.1 Test datasets

For the evaluation of the models, we obtained benign domain names from rankings of the most popular visited websites such as Alexa, Cisco Umbrella, and Quantcast. The malware-generated domain names were collected in the sandbox of an operational environment. We train and test both models on the same datasets. The binary classification dataset contains 13 400 training samples and 6 600 test samples, evenly balanced between DGA and benign domains. The same binary classification models are further tested on two recent DGAs: FluBot and Sunburst. Domains from these malware families were not included in the training dataset, so are completely unknown to the model. For the FluBot DGA, we use an evaluation set of 10 000 samples, whilst for the Sunburst DGA we had at the time (March-April 2021) 5 000 domain names available.

The dataset used for the family classification task consists of 32 880 training samples and 16 195 testing samples from 77 malware families. Figure 2 shows the number of samples per family in the training and test set. We observe that the dataset is not evenly balanced, as 21 out of 77 families (around the 27% of the families) account for less than 100 samples in the training set. The rarest class in the dataset is `gozi3m`, which only has one sample in the training set.

Table 1: Results of binary classification (Support = 6600)

Method	Accuracy	Precision	Recall	F1-score
Feature-based	0.9286	0.9287	0.9286	0.9286
Featureless	0.9565	0.9570	0.9564	0.9565

Table 2: Classification results on the FluBot and Sunburst malware domain names

Evaluation set	Accuracy	
	Feature-based	Featureless
FluBot	0.986	0.993
Sunburst (full domain)	0.0000	0.0002
Sunburst (subdomains)	0.831	1.0000

4.2 Binary classification results

To test the feature-based approach, we feed a RF model with the features defined in Section 3.1. We perform a 10-fold cross validation to reduce overfitting, together with a grid search to find the best `max_depth` and `n_estimators` combination for the model (i.e. `max_depth` = 20 and `n_estimators` = 100).

The binary featureless classifier consists of:

- one embedding layer that projects the sequences into a higher dimension space to allow for more degrees of freedom;
- one TCN residual block with 8 filters, a filter size $k = 4$, and 6 dilated convolutions $d = [1, 2, 4, 8, 16, 32]$;
- one output layer, which predicts the final class based on a sigmoid function.

The output for the binary classification is either 0 (non-DGA) or 1 (DGA). We measure the performance of both classifiers in terms of Accuracy, Precision, Recall and F1-Score [11]. The metrics for the first binary classification task are shown in Table 1. We observe that both methodologies overcome 0.9 in all the considered measures. The featureless model provides better detection capabilities, reaching 0.9565 in accuracy.

Table 2 reports the binary classification results for FluBot and Sunburst. The FluBot DGA is detected by both approaches with high accuracy, even though it is completely unknown to the model. For the featureless approach, we noticed that the few domain names that were misclassified contain more vowels (median = 6.0, mean = 6.05, std = 1.26) compared to the rest (median = 4.0, mean = 3.73, std = 1.60) and contain one or more proper words (e.g. `sitchromsvkghal.cn`, `almanarnebttaic.cn`, or `freasomedpmfitn.com`).

The Sunburst case was far more challenging. When using the full domain name for labelling, we obtained very poor classification results. The only domain classified correctly in this process was unusually long. This poor classification result can be explained by the fact that in Sunburst-generated domains only the first subdomain

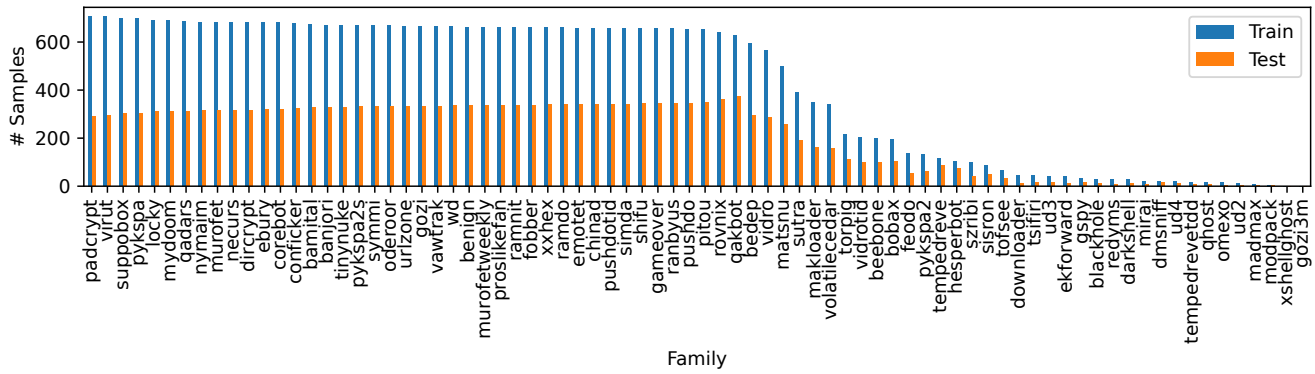


Figure 2: Number of samples per family in the training set

is obviously suspicious, whereas the following three or four subdomains look rather benign (e.g. `03l3eootsf8ifn9dhu4tslosf1ja0l2b.appsync-api.us-west-2.avsvmcloud.com`). The first subdomain is not randomly generated but instead encodes information about the infected computer. This domain name structure is difficult to classify. For this particular DGA, we should apply a different labelling procedure: first split the original domain into subdomains, label each subdomain individually and, if at least one of the subdomains is detected as DGA-generated, label the full domain name as such. With this procedure, we are able to significantly improve the performance of both approaches, achieving 100% accuracy on the Sunburst evaluation set in the featureless case. This finding suggests that applying classifiers to both the full domain name, and the extracted subdomains may increase the detection rate.

4.3 Family classification results

Table 3: Results of family classification

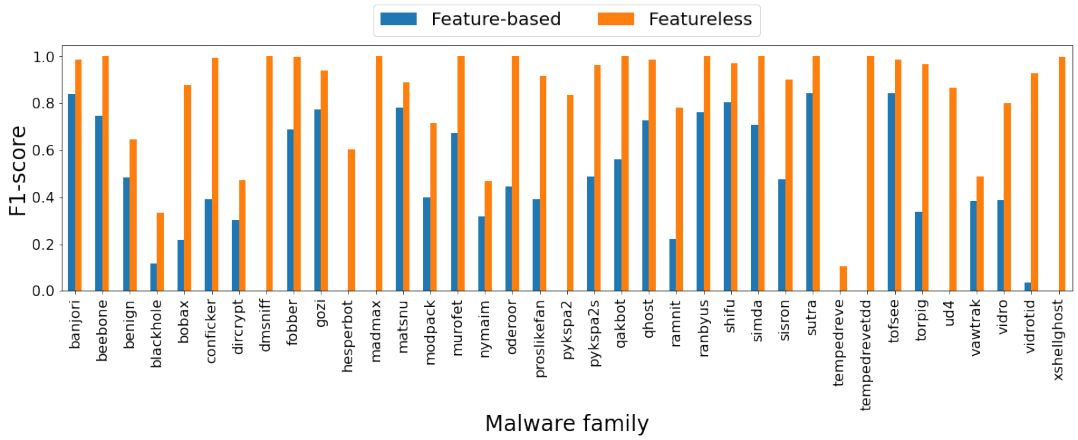
Averaging	Method	Precision	Recall	F1-score
macro avg	Feature-based	0.6709	0.6403	0.6421
	Featureless	0.7730	0.7880	0.7761
weighted avg	Feature-based	0.7283	0.7233	0.7126
	Featureless	0.8021	0.8092	0.8035

The performance of the approaches for DGA family classification has been evaluated. Table 3 reports the macro and weighted average measures for both approaches. Also, in this case, the featureless method outperforms the feature-based approach by 0.10 to 0.17 points on average. Figure 3 reports the detailed F1-Scores for some of the most interesting families in the dataset. Figure 3b shows the F1-scores of malware families on which the featureless approach achieves at least a 0.1 improvement over the feature-based approach. This is the case for 37 classes. In some cases, such as 'dmsniff', 'madmax', 'tempedrevetdd', and 'xshellghost', the difference can be as large as 0.9. However, we note that the support in the training set is also small, with less than 20 samples for each of these four malware families. There are a 17 classes for which the feature-based approach achieves better classification, as can be seen in Figure 3a. For these specific malware families, the RF classifier is able to slightly outperform the multi-class TCN, the largest difference being a 90% improvement in the F1-score for the 'gspy' and 'wd4' families. In particular, these malware families generate domains that contain many digits, an aspect which is easier to model with an explicit feature. The overall better performance of the featureless approach can be explained by the fact that the deep learning algorithm can extract complex patterns that are more representative of the structure of the domain names, in comparison to the features employed by the Random Forest classifier. Moreover, this experiment suggests that a combination of these two approaches may lead to an overall better classification.

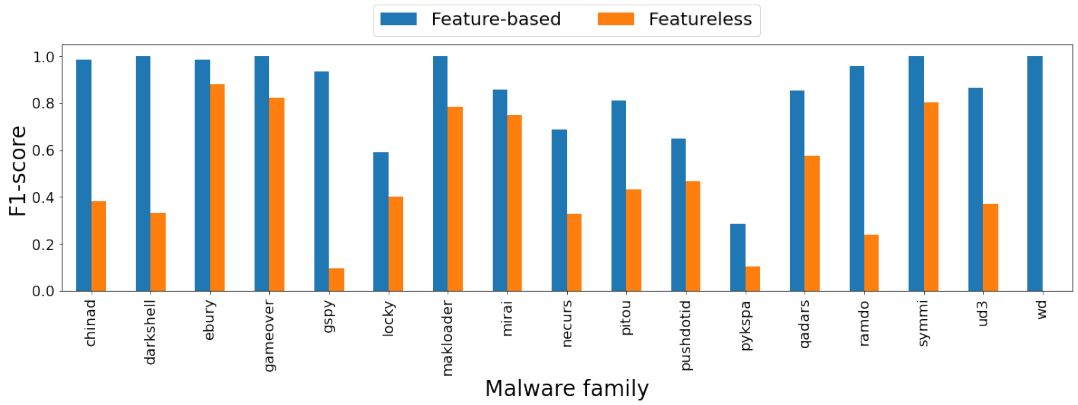
4.4 Discussion

We have observed that both methods show good DGA detection capabilities, with some limitations due to the specific structure of some of the analysed domains (e.g. those belonging to the Sunburst family). This problem can be overcome by applying the subdomain classification strategy described in Section 4.2, wherein we showed that the subdomain classification significantly improves the detection capabilities in the presence of complex domain name structures, especially in the featureless case.

Malware family classification proved to be a harder task. As already mentioned in Section 4.3, the performance of both models are heavily dependent on the number of samples available for each family. The worst scores are yielded in case of under-represented families, as well as families showing a very variable structure. In many cases, samples coming from under-represented families are assigned to larger classes having a similar structure (e.g. using the same TLD or having a comparable domain length). We further showed that, although the featureless approach achieves better classification results on the majority of families, several classes benefit from the feature-based approach. Moreover, when inspecting misclassified samples from families such as 'ud4' and 'dmsniff', we discover new distinguishing patterns, such as the first two characters or the inclusion of a specific letter ('ud4' domains do not contain the letter 'b'), which can be included in the feature set. These findings indicate that combining the two approaches, either in the form of an ensemble classifier [29] or a multi-input neural



(a) Malware families on which the featureless approach performs better



(b) Malware families on which the feature-based approach performs better

Figure 3: F1-Score for selected DGA families.

network, could achieve higher accuracy in the family classification task.

The featureless approach entails training a deep learning model and requires appropriate computational resources. All the results for the TCN binary and family classifiers were obtained using a spot virtual machine (VM) in Microsoft Azure. Specifically, we employed one NCv3 VM with 112 GB of memory and one GPU (V100 card). The training process was quite rapid, with the family classifier learning a sufficient representation of the domains in less than ten minutes. In the feature-based case, while the model training and parameter tuning were relatively fast tasks, requiring less than ten minutes as well, the most computationally intensive operation was the feature extraction, and the pairwise string distance in particular. The entire process ran on a local machine, equipped with one Intel® Core® i7 processor, providing 4 cores and 32 GB of RAM. This setup strongly impacted the required computation time, that was growing up to days, forcing us to limit the dataset size. Distributing the distance matrix computation [6], or relying on more scalable techniques [16] may be a viable way to address this issue.

4.5 Operational Experience

The featureless approach has been open-sourced and is available on Github[2]. Since the start of 2021, it has also been tested operationally in the production environment of the Shadowserver Foundation[3], with additional Dashboard functionality added to support improved Threat Hunting. In mid-2021, the tool led to the identification of new DGAs for the 'phorpiex' and 'mydoom' threats, and also 'm0yv/expiro' (which was also discovered and reported independently by other parties), and enabled sinkholing[4] of threats, such as 'Phorpiex', 'M0yv' and 'Neksminer', and sharing data on infected machines with the Internet Defender community. The Shadowserver Foundation exposes an API that can be used to perform queries using the implemented model. For example, it can be used to determine whether a provided domain name is malicious and return a list of classified DGA-generated domains, given the hash of a malware sample.

[2]https://github.com/COSSAS/dgad
[3]https://www.shadowserver.org/
[4]Sinkholing is a technique whereby a resource used by malicious actors is redirected to a benign listener that can monitor connections from infected devices.

5 CONCLUSION

In this paper, we examined the applicability of AI-based methods to DGA detection and classification for network security. To this end, we proposed two different approaches, a RF classifier employing string-based features, and a TCN that uses only a numerical encoding of domain names as input. When comparing the two approaches on the binary classification task, we observed that the featureless approach slightly outperforms the feature-based classifier, with approximately 3% increase in accuracy. The multi-class evaluation showed a drop in the classification performance of both models, largely due to the class imbalance present in the dataset. Moreover it was also observed that some malware families can be better identified with one of the methods in particular, a fact which can be linked with their specific DGA characteristics.

One of the biggest challenges highlighted by this study was the problem of class imbalance. A natural progression of our work is to analyse how this can be tackled with cost sensitive training, resampling techniques, or with more recent approaches such as maximum class separation [13]. The models' power to generalize is also an aspect worth further investigation. In practice, DGA classifiers should be able to detect domain names that are generated from new malware, with unknown algorithms and seeds. To this end, one could define strict confidence thresholds for each malware family and label new samples when these thresholds are not met.

ACKNOWLEDGMENTS

The research leading to this publication was supported by the EU H2020 project SOCCRATES (833481). We kindly thank Piotr Kijewski from the Shadowserver Foundation for the datasets and support provided in the project, and Federico Falconieri from TNO for improving and publishing our code.

REFERENCES

[1] Almashhadani, A., Kaiiali, M., Carlin, D., and Sezer, S. Maldomdetector: A system for detecting algorithmically generated domain names with machine learning. *Computers & Security* (03 2020), 101787.

[2] Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv abs/1803.01271* (2018).

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

[4] Drichel, A., Meyer, U., Schüppen, S., and Teubert, D. Analyzing the real-world applicability of DGA classifiers. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (2020), ARES '20.

[5] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (1996), vol. 96, pp. 226–231.

[6] Faroughi, A., Javidan, R., Mellia, M., Morichetta, A., Soro, F., and Trevisan, M. Achieving horizontal scalability in density-based clustering for urls. In *2018 IEEE International Conference on Big Data (Big Data)* (2018), IEEE, pp. 3841–3846.

[7] Fireeye. Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor. https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html.

[8] Fraunhofer FKIE. CryptoLocker. https://malpedia.caad.fkie.fraunhofer.de/details/win.cryptolocker.

[9] Fraunhofer FKIE. FluBot. https://malpedia.caad.fkie.fraunhofer.de/details/apk.flubot.

[10] Fraunhofer FKIE. Gameover P2P. https://malpedia.caad.fkie.fraunhofer.de/details/win.gameover_p2p.

[11] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[12] Hoang, X. D., and Vu, X. H. An improved model for detecting dga botnets using random forest algorithm. *Information Security Journal: A Global Perspective* (2021), 1–10.

[13] Kasarla, T., Burghouts, G. J., van Spengler, M., van der Pol, E., Cucchiara, R., and Mettes, P. Maximum class separation as inductive bias in one matrix, 2022.

[14] Li, Y., Xiong, K., Chin, T., and Hu, C. A machine learning framework for domain generation algorithm-based malware detection. *IEEE Access 7* (2019), 32765–32782.

[15] Liang, J., Chen, S., Wei, Z., Zhao, S., and Zhao, W. Hagdetector: Heterogeneous dga domain name detection model. *Computers & Security 120* (2022), 102803.

[16] McInnes, L., Healy, J., and Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw. 2*, 11 (2017), 205.

[17] Saxe, J., and Berlin, K. expose: A character-level convolutional neural network with embeddings for detecting malicious urls, file paths and registry keys. *ArXiv abs/1702.08568* (2017).

[18] Schüppen, S., Teubert, D., Herrmann, P., and Meyer, U. FANCI : Feature-based Automated NXDomain Classification and Intelligence. In *USENIX Security Symposium* (2018).

[19] Selvi, J., Rodríguez, R. J., and Soria-Olivas, E. Detection of algorithmically generated malicious domain names using masked n-grams. *Expert Systems with Applications 124* (2019), 156–163.

[20] Sivaguru, R., Choudhary, C., Yu, B., Tymchenko, V., Nascimento, A., and Cock, M. D. An evaluation of dga classifiers. *2018 IEEE International Conference on Big Data (Big Data)* (2018), 5058–5067.

[21] Sivaguru, R., Peck, J., Olumofin, F. G., Nascimento, A. C. A., and Cock, M. D. Inline detection of DGA domains using side information. *CoRR abs/2003.05703* (2020).

[22] Soleymani, A., and Arabgol, F. A novel approach for detecting dga-based botnets in dns queries using machine learning techniques. *Journal of Computer Networks and Communications 2021* (2021).

[23] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499* (2016).

[24] Vosoughi, S., Vijayaraghavan, P., and Roy, D. Tweet2Vec: Learning Tweet Embeddings using Character-level CNN-LSTM Encoder-Decoder. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (2016).

[25] Vranken, H., and Alizadeh, H. Detection of dga-generated domain names with tf-idf. *Electronics 11*, 3 (2022), 414.

[26] Woodbridge, J., Anderson, H. S., Ahuja, A., and Grant, D. Predicting domain generation algorithms with Long Short-Term Memory Networks, 2016.

[27] Yu, B., Pan, J., Hu, J., Nascimento, A., and De Cock, M. Character level based detection of DGA domain names. In *2018 International Joint Conference on Neural Networks (IJCNN)* (2018), pp. 1–8.

[28] Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Cambridge, MA, USA, 2015), NIPS'15, MIT Press, p. 649–657.

[29] Zhou, Y., Cheng, G., Jiang, S., and Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks 174* (2020), 107247.