# DOLPHIN: Phonics based Detection of DGA Domain Names

Dan Zhao*†, Hao Li*, Xiuwen Sun‡, and Yazhe Tang*§
*School of Computer Science and Technology, Xi'an Jiaotong University
† Xi'an University of Finance and Economics
‡Anhui University
§Science and Technology on Communication Networks Laboratory
Email: {zhaodan.echo, mr.xiuwen}@gmail.com, hao.li@xjtu.edu.cn, yztang@mail.xjtu.edu.cn

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685325

***Abstract*—Botnets are the machines that increasingly controlled by cybercriminals to perform various attacks. They use Domain Generation Algorithm (DGA) to frequently generate their illegitimate domains for preventing detection. To overcome such dynamics, existing solutions try to capture the characteristics of domain names, such that the automatically generated domains can be identified. However, those solutions are not conformed to the linguistic conventions of reading and writing. For a comprehensive understanding of strings of domain names, we present DOmain Linguistic PHonIcs detectioN (DOLPHIN), a novel method that can detect the illegitimate domain names generated by DGAs. Considering the correspondence between pronunciations and spellings, we design the DOLPHIN patterns. They are the classification of vowels and consonants in variable lengths as follow the principles of phonics. DOLPHIN recognizes strings of domain names and reconstructs them with the components of variable-length vowels and consonants following the DOLPHIN patterns. We implement the features used DOLPHIN in supervised learning methods and compare them to the foremost method FANCI. Experimental results show that, compared to FANCI with RFs, DOLPHIN can achieve higher detection accuracy of 0.0238 in average with lower FPR without much overhead.**

***Index Terms*—DGA, Domain Names, Security, RFs, Botnets, Machine Learning**

## I. Introduction

Botnets are distributed networks that consist of infected devices (bots), including computers, cellphones, and Internet of Things devices, which can launch various attacks, e.g., DDoS, data stealing and spam sending. They may make target networks or hosts inaccessible or crashed. It is thus important to block the botnets for preventing from the attacks.

One mainstream way to achieve this goal is to identify and then block the communication channel between bots and Command-and-control (C&C) servers [1]. For example, since many IP addresses or domain names of C&C servers are hard coded into malware binaries, one can undertake reverse engineering of binary and maintain large blacklists of those IPs or domain names [2]. Then, online security systems can block the connections to those IPs or domains according to the blacklists. Unfortunately, this method becomes ineffective these days, as botnets tend to use Domain Generation Algorithm (DGA) [3] to periodically generate a large number of pseudo-random domain names. These illegitimate domain names, known as the Algorithmically Generated Domains (AGDs) [4], can change very frequently. Only few of them are registered. Thus, they can be barely identified by fixed blacklists. DGA has become the cornerstone technique for botnets. Specifically, measurements indicate that the majority of the observed botnets used DGA as their only mechanism for communication. Most of them are valid for a short period, e.g., within only one day [5]. Besides, they may spread out over different top-level domains worldwide. Hence, identification of AGDs is the key to blocking most botnets.

Many attempts have been made to identify the AGDs. The mainstream method uses machine learning techniques to analyze the patterns in strings of domain names: a legitimate domain must be meaningful to human; otherwise, it tends to be an AGD [6]–[9]. The major features they are concerned with include structural features (e.g., the lengths of domain names, the numbers of subdomains) and statistical features (e.g., n-gram, entropy) [6] [8]. These features are used in machine learning classifiers. They are the crucial inputs to the results of identification since the classifiers learn from these inputs and give the mapping between them and the classified results. More recent approaches [6] claim that involving linguistic features (e.g., ratio of vowel, ratio of consecutive consonants) can boost the accuracy of the detection of AGDs.

However, previous linguistic features might not well capture the real semantics carried by a domain, which may bring false positives and further lower the accuracy rate. Take the domain name `nationalgeographic.com` as an example. The classical methods would tend to mark it as an AGD, since they identify this domain has many repeated characters (5 in total) and consecutive consonants (`lg`, `gr` and `ph`, 6 in total). Obviously, this is a false positive since this domain name is the official website for the famous magazine: National Geographic Magazine.

We observe that the root cause of the above false positive is the imprecise classification of vowels and consonants. The classical methods classify them based on single letters, i.e., `a`, `e`, `i`, `o` and `u` are classified as vowels and other letters are viewed as consonants. However, the classification is not precise, as vowels and consonants are actually defined by their pronunciations in words. So, the single characters may not fully capture components in words. For example, `graphic` pronounces /græfık/, which contains two vowels, i.e., /æ/

978-1-7281-8104-2/21/$31.00 ©2021 IEEE

and /ɪ/, and three consonants, i.e., /gr/, /f/ and /k/. It is obvious the pronunciation of /f/ maps to the spelling of ph. As a result, ph should be considered as a single consonant, instead of two consecutive ones. The spellings of ph, i and c are called graphemes, i.e., one or multiple letters, which are the smallest units of spellings [10]. The rationale behind the above examples is phonics which proves that graphemes can be spelled in variable lengths and gives the mapping between the graphemes and their pronunciation [10]–[12]. Therefore, it is necessary to change the classification of vowels and consonants by following the principles of phonics, so that linguistic features can better contribute to the detection of AGDs.

In this paper, we propose DOmain Linguistic PHonIcs detectioN (DOLPHIN). We analyze and define the DOLPHIN patterns, which present a novel classification of vowels and consonants. The patterns comprehend the correspondence between graphemes and their pronunciations. Then, DOLPHIN adopts such varied-length patterns to extract the linguistic features from domains in a more precise way. In the former example of nationalgeographic.com, DOLPHIN views a, ion, al, e, o, i as vowels, and n, t, g gr, ph, c as consonants. Since there is no consecutive consonant and only one repeated character (i.e., a) with the current classification, the domain can be correctly identified as a normal one.

As far as we know, DOLPHIN is the first to introduce phonics to detecting AGDs. Specifically, we propose the DOLPHIN patterns which classify vowels and consonants following the principles of phonics i.e., they can map to single letters or small sequences of letters. We also provide a concrete implementation of DOLPHIN and evaluate it with real configurations. DOLPHIN computes linguistic features applying the DOLPHIN patterns, which are fed to different classifiers. The results show lower false positive rate and higher accuracy compared to the state-of-the-art approaches.

The reminder of this paper is organized as follows. Section II describes the patterns and phonics-based features. We outline the implementations in Section III. The evaluation and the analysis of the experiments are described in Section IV. The related works are introduced in Section V. Finally, we conclude this paper in Section VI.

## II. Design of DOLPHIN

This section presents the design of DOLPHIN. We first introduce the DOLPHIN patterns that classify vowels and consonants based on phonics. Then we show how to use DOLPHIN patterns in the linguistic features.

### A. DOLPHIN Patterns

Vowels and consonants are defined by their pronunciations, and phonics connects the pronunciations with their spellings. The key to phonics is that the pronunciation of a vowel or consonant maps to a grapheme, which can be a single letter or a multigraph that consists of multiple letters. Phonics enables the ability to precisely classify vowels and consonants by identifying the spellings of graphemes in a word. Specifically, we summarize the DOLPHIN patterns based on the classical principles of phonics [10], [11], [13], a mapping from the spellings of graphemes to their classification, i.e., vowels or consonants.

We name the new types of vowels and consonants D-vowel and D-consonant respectively, shown in TABLE I. D-vowels consist of 5 vowel characters, 27 vowel digraphs (i.e., multigraphs with 2 letters) and 11 vowel trigraphs (i.e., multigraphs with 3 letters). In a similar way, D-consonants consist of 21 consonant characters, 43 consonants digraphs and 11 consonant trigraphs. For example, in the DOLPHIN patterns, er in butter is a vowel digraph by our new classification, instead of the vowel e and the consonant r. We exclude the multigraphs with more than 3 letters, because they are infrequent.

TABLE I
DOLPHIN Patterns

| Type | Length | Graphemes |
|---|---|---|
| D-Vowel | *1* | a,e,i,o,u |
| | *2* | ai,al,ar,au,aw,ay,ea<br>ee,ei,er,eu,ew,ey,<br>ia,ie,ir,oa,oe,oi,oo<br>or,ou,ow,oy,ue,ui,ur |
| | *3* | air,ear,eer,igh,ign,ing<br>ion,oew,ore,our,ure |
| D-Consonant | *1* | b,c,d,f,g,h,j,k,l,m,n<br>p,q,r,s,t,v,w,x,y,z |
| | *2* | bl,br,ch,ck,cl,cr<br>dr,fl,fr,gh,gl,<br>gr,kn,ld,lk,mb,mn<br>mp,nd,ng,nk,nt,<br>ph,pl,pn,pr,ps,qe<br>qu,rh,sc,sh,sk,<br>sl,sm,sn,sp,st,sw<br>th,tr,wh,wr |
| | *3* | dge,gue,nch,que,shr,spl<br>spr,squ,str,tch,thr |

Unlike traditional detection system's classification of vowels and consonants, the phonics-based vowels and consonants behave as the letters of different lengths to represent vowel or consonant sounds. For example, in the DOLPHIN patterns, the lengths of the graphemes of u and er in butter is 1 and 2 respectively.

### B. Phonics-based Features

Detecting AGDs using phonics is now considered as classifying domain names by the features extracted from the underlying characteristics of the graphemes. The proposed DOLPHIN patterns can be used in most linguistic features, since most of them are computed by the occurrences of vowels and consonants. We choose 3 representative linguistic features to analyze: the vowel ratio, the ratio of repeated characters and the ratio of consecutive consonants [6] [8].

The vowel ratio is the calculated as the ratio of the number of vowel characters to the length of the dot-free

public-suffix-free domain (i.e., a domain name ignoring separating dots and its valid public suffix) [6]. For example, `hostfacebook` is the dot-free public-suffix-free domain of `host.facebook.com`. And `facebook` is the dot-free public-suffix-free domain of `facebook.com`. The vowel ratio of `facebook.com` for the character-based methods' results in 4/8 for `facebook` has 4 vowel characters. While DOLPHIN yields 3/7 for it has 3 vowels and considers the `oo` as a whole part.

The ratio of repeated characters is referred as the ratio of the number of characters that repeated throughout a public-suffix-free domain to the number of characters appeared in the dot-free public-suffix-free domain. In the previous example, this feature evaluates to 1/7 in the character-based methods. By contrasts, the value is sparkly 0 under the condition of DOLPHIN.

The last feature, ratio of consecutive consonants, is defined as the sum length of several successions (whose length is greater than or equal to 2) of consonant characters divides the length of the dot-free public-suffix-free domain. In the example of `google`, ratio of consecutive consonants of the character-based methods calculates as 2/6. And that value of DOLPHIN calculates as 0. Because `gl` is seem as a whole part and consequently there is no consecutive consonants.

There are some theoretical characteristics of the features based on the DOLPHIN patterns. A legitimate domain trends to have a larger vowel ratio, a smaller ratio of repeated char and consecutive consonant. On the contrary, an AGD has a fairly high probability of a smaller vowel ratio, a larger ratio of repeated chars and consecutive consonant.

## III. Implementation

To proof the effectiveness, we implement DOLPHIN and the phonics-based features with machine learning techniques in the detection of DGA generated domain names from legitimate domain names.

We present the overview of the implementation and the way we use DOLPHIN, shown in Fig. 1. We employ supervised learning techniques to detect AGDs. According to the order of processing steps, it is made up with the preprocessing module, DOLPHIN, the feature extraction module, the training module, and the classification module.

In the preprocessing step, negative data are cleaned though deleting the samples without any public suffix. Because in open data, a domain name that don't end with a public suffix is unacceptable for DNS system. Usually, these domain names are produced by mistyping or misconfiguration, so they are not actually legitimate domains.

After that, DOLPHIN addresses the preprocessed data and yields the public-suffix-free names, and next reconstructs the D-domain names follow the DOLPHIN patterns.

Then we extract the linguistic features, structural features, and statistical features from domain names in the extraction module. The structural features and the statistical features are used as the same as other methods, like the lengths of domain names, the numbers of subdomains, entropy. They

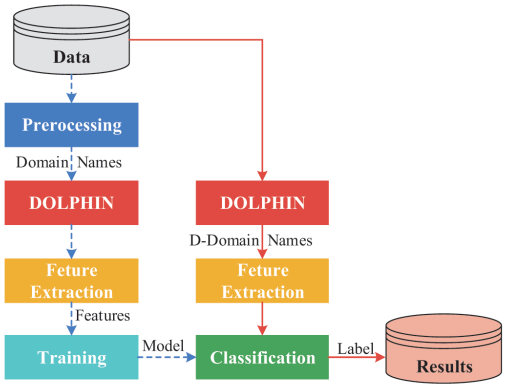


Fig. 1. Implementation of Detection of AGDs.

are straightforward extracted from domain names. The three linguistic features which are related to vowel and consonant letters, are extracted from the D-domain names.

Next, in the training module, the features are accepted and trained by a classifier and it will yield a trained model.

At last, in the classification module, the trained model reads a submitted domain name. It then extracted features from the corresponding D-Domain name, and then predicts whether the domain is an AGD or not. The classification model can finally be an application that detecting DGA generated names in real-time by the trained model.

## IV. Evaluation

In this section, we evaluate DOLPHIN. The experiments are performed over a series of datasets and aim to answer the following questions:

*(1) Can DOLPHIN achieve higher accuracy and other metrics compared to the state-of-the-art approach?* We find that DOLPHIN indicates a better performance than FANCI [6] which is one of the most advanced approaches in terms of each evaluation metric on different sizes of datasets. With DOLPHIN employed, the overall mean ACC of 5-folds increases to 0.9384 (by 0.0265) on different datasets. Especially, FPR is reduced by 0.0220 (28.76%).

*(2) Can DOLPHIN generalize to other classifiers?* Experiments show that DOLPHIN with GXBoost classifier and SVMs also achieve the overall mean accuracy of 0.9322, 0.9229 respectively. They are 2.23% and 1.21% more accurate than FANCI with RFs.

*(3) Does DOLPHIN bring other overhead for better accuracy?* Experiments show that the training time of two methods using RFs are within the same order of magnitude. But DOLPHIN shows a better performance.

### A. Experimental Setup

To ensure that the performance of DOLPHIN is brought by the linguistic features applying the DOLPHIN patterns, we use the same features as FANCI's. So, with a given specific classifier, the only difference between two implementations is the extraction of the 3 linguistic features: DOLPHIN's features

are extracted from the D-domain names and FANCI's are extracted from the original domain names.

**Data Set.** DOLPHIN and FANCI are supervised methods, which require the labeled data. We get positive samples, i.e., DGA generated domain names, from OSINT DGA feed [14] and negative samples, i.e., legitimate domain names, from Alexa Top domain names. We have 40 datasets with the sizes ranging from 500 to 20000. The numbers of negative samples and positive samples are the same in each dataset.

**Experimental Design.** To evaluate the proposed method, we perform two series experiments.

First, models trained with RFs are performed on different sizes of datasets. For comparison, 3 groups of features are extracted. The first group is the features of DOLPHIN. The second is the features of FANCI. The third group of features without the 3 linguistic features is used as a baseline. The baseline is a representation for those methods that do not use linguistic features. The difference among them is that the baseline has 18 features, while FANCI has 3 extra features based on single characters of vowels and consonants and DOLPHIN upgrades the 3 features based on D-Vowels and D-Consonants. These groups of features are trained, and their output models will be used in the prediction of the same AGDs respectively. They would measure the effectiveness of DOLPHIN in detecting AGDs, if any, compared with the characters of vowels and consonants-based features.

Second, the RFs, SVMs and XGBoost are compared on different sizes of datasets. This is to demonstrate whether DOLPHIN can generalize to other classifiers except for RFs.

Each experiment is carried out with a 5-fold cross validation (CV) for less biased estimates. It means that a dataset is randomly silted into 5 groups and every group will be used as hold-out data with the remaining data as training data.

All experiments are performed at an x86 PC with 6×Intel 3GHz CPU and 8G RAM on Windows 10. We do not use a high-end server because the focus of the experiments is the accuracy instead of the speed.

**Evaluation Metrics.** To measure the quality of the methods, we introduce types of evaluation metrics. Accuracy, FNR and FPR are used to compare the performance of DOLPHIN to FANCI's. Accuracy is defined as $ACC = \frac{TP+TF}{TP+TF+FP+FN}$, and measures the ratio of the number of correct predictions to the total number of samples. Here $TP$ denotes the number of AGDs that are correctly predicted. $FN$ denotes the number of AGDs that are predicted to legitimate domains. $TN$ denotes the number of legitimate domains that are correctly predicted. $FP$ denotes the number of legitimate domains that are predicted to ADGs. False Positive Rate (FPR) is defined as $\frac{FP}{TN+FP}$. In the problem of detecting AGDs, it means the proportion of legitimate domains that are predicted to ADGs of all the legitimate domains.

Among metrics, FPR gains more attention in practical applications in terms of user experience. A Higher FPR may trigger off numbers of false alarms.

### *B. Different Features with RF Model*

The presentation of the mean accuracy of the 5-fold results using RFs with DOLPHIN, FANCI and the baseline are shown in Fig. 2.

Fig. 2 shows that DOLPHIN and FANCI both have larger ACC compared with the baseline. Precisely, the overall mean ACC of the two methods rise by 0.0628 (7.17%) and 0.0363 (4.15%) respectively. It illustrates that the application of vowels and consonants in linguistic features are helpful. The overall mean ACC of DOLPHIN is 0.9384, which has a increase of 0.0265 (2.91%) compared with FANCI. It reveals the validity of the DOLPHIN patterns, i.e., the graphemes could catch more accurate characteristics of domain names than the traditional methods. The ACC of the above two methods tend to stabilize when the size of samples increases to 3000. Note that the ACC of DOLPHIN is still greater in small datsets, e.g., at the size of 2000, where the DOLPHIN's mean ACC is 0.9450.

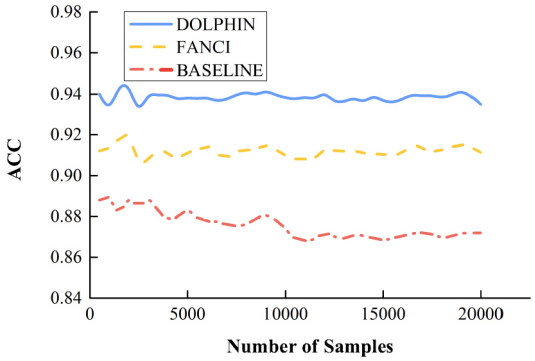


Fig. 2. ACC of DOLPHIN, FANCI and baseline on Different Sizes of Samples

We also compare FPR among DOLPHIN, FANCI and the baseline. The results are shown in Fig. 3. It shows that DOLPHIN performs best, and the baseline performs worst. Specifically, the overall mean FPR for them are 0.0545, 0.0765 and 0.1142 respectively over the whole datasets. The overall mean FPR of DOLPHIN presents a reduction of 0.0220 (28.76%) compared to FANCI. As the number of samples grows, the FPR of DOLPHIN and FANCI are more stable, but that of the baseline is growing. It means that it is more likely for the baseline to predict legal domains as AGDs mistakenly, especially using larger sizes of datasets.

The performance of FANCI in our experiments is lower than that in its original paper [6], majorly due to the different datasets we use. The domain names of FANCI come from traffic which contain host names, e.g., `www.google.com`. While our experiments use the domain names without host names, e.g., `google.com`. Few features are useless in our data, e.g., the feature that whether a domain name has `www` as a prefix, thus leads to a lower accuracy.

DOLPHIN can classify the domains correctly by changing the values. Because the values of the AGDs' features mainly differ from those of the legitimate domains. AGDs are gen-

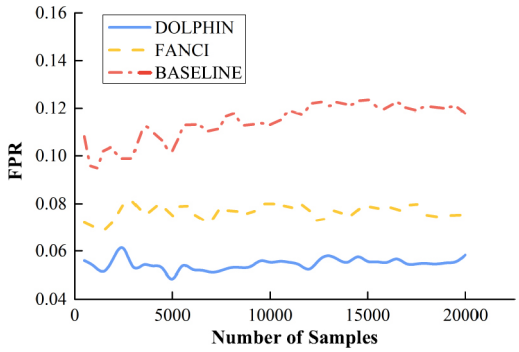


Fig. 3. FPR of DOLPHIN, FANCI and baseline on Different Sizes of Samples

erated by different kinds of algorithms, hence they are of different characteristics which cannot easily be summarized just by one pattern. However, the structures of the legitimate domain strings are relatively describable. They result in some specific values on behalf of the structures. So, the domains that having the values of features in the opposite manners can be identified as AGDs.

### C. Different models with Phonics Features

The ACC of DOLPHIN with RFs, XGBoost and SVMs appear as shown in Fig. 4. The ACC of RFs and GXBoost remain relatively stable with the overall mean values of 0.9384, 0.9322 respectively. The overall mean ACC of SVMs is 0.9229. But the ACC of SVMs is consistently falling as the number of samples arises. It might be due to noises in the input datasets. While the overall mean ACC for DOLPHIN using different methods are larger than that of FANCI's best mean result (i.e., employed RFs). DOLPHIN with the different classifiers are 2.91% 2.23% and 1.21% accurate than FANCI with RFs, respectively.

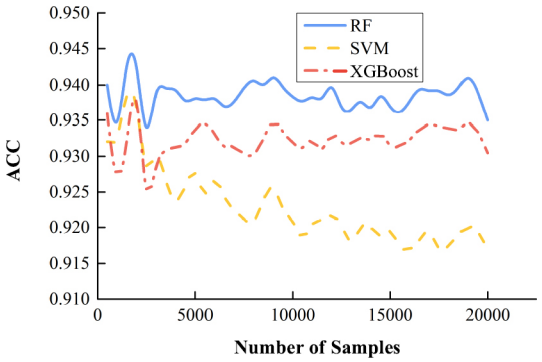


Fig. 4. ACC of DOLPHIN Using Different Classifiers on Different Sizes of Samples

We compare the FPR of DOLPHIN using the different classifiers, shown in Fig. 5. The overall mean FPR of SVMs and XGBoost is 0.0665, 0.0604. The two values are smaller than that of FANCI using RFs. On the small datasets, i.e., numbers of samples for the datasets are from 500 to 2500, SVMs show the smallest FPR. And XGBoost presents a greater FPR among them from the datasets of 500 to 4000. On the datasets whose number of samples are more than 2500, RFs perform with the lowest FPR. It means that if we prefer less legitimate domains which are mistakenly predicted, SVMs are a smart choice on the condition of using small datasets. And RFs are better on the condition of using more training samples.

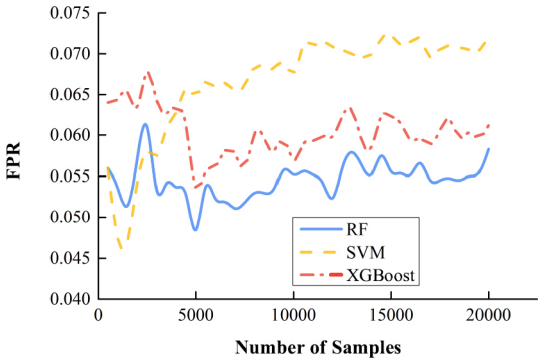


Fig. 5. FPR of DOLPHIN Using Different Classifiers on Different Sizes of Samples

In the experiment, we learn that the performance of RFs is the best in general. They have the largest ACC and the smallest FPR among the methods on most datasets. Meanwhile, SVMs work slightly worse than the others. Especially, they indicate performance degradation since the dataset size increases to 10000. We find that the metrics of DOLPHIN employed SVMs are even better than that of FANCI employed RFs. So, we believe that DOLPHIN can perform well with the change of classifiers.

### D. Training Speed

The time overhead for DOLPHIN is negligible compared to FANCI. On the dataset of 10000 samples in which 80% training data, it takes DOLPHIN 1.06s for training, which is 0.03s more than that of FANCI.

In our experiments, the linguistic features using the design of D-vowels and D-consonants in DOLPHIN is by far the most important single variable in determining the improvement of the performance.

## V. Related Work on AGD Detection

### A. Traditional machine learning approaches

Most machine learning detection methods classify domains with manually created features and machine learning models and can be divided into two categories. The first category is string-based methods. Pleiades [9] is proposed to cluster domains by statistical features and bipartite graphs using Hidden Markov Models (HMMs) and then classify domains using NXDomian responses traffic. FANCI [6] is the state of art system that proposed classification system using machine learning models and 21 meaningful features extracted by

domains. It uses data obtained from NXdomain. This is the closest work to ours. The primary difference is that DOLPHIN implements the linguistic features with phonics introduced, which can capture more characters of domain names.

The second category is time-based that employs chronological features of DNS transactions, e.g., the time series and gaps between the DNS request and response [15], [16]. BotFinder [15] uses a machine leaning model and offers the information extracted from the reassembled Netflow and traces. But it has to get sequences of chronologically-ordered flows first. PsyBoG [16] leverages the frequencies of botnet behaviors to distinguish them from the normal behaviors applied the power spectral density (PSD) analysis. Phoenix [8] first uses a combination of IP pools, linguistic features of domain names to cluster and identify AGDs. But the only linguistic feature used is n-gram normality score which measures the relation within characters spitted by the constant lengths. That is not entirely accord with the theory of linguistics. While these approaches could be effective in dynamic environments, they cannot be used in real-time, as they have to collect DNS traces beforehand. In contrast, the patterns extracted in the string-based methods can be directly used in an on-line system, e.g., an intrusion detection system [6], to identify and block C&C channels.

### B. Deep learning approaches

Some approaches use deep learning techniques, e.g., neural network (NN), to identify AGDs from the normal domain names. In [7], LSTM network based method is first introduced in detecting DGAs. The approach only uses strings of domain names as input without any features extraction. It can classify 90% AGDs with a false positive rate of 1:10000. It only takes 20 ms for predicting a domain name. Bin [17] compared convolutional and recurrent neural networks (CNN and LSTM respectively) on Non-Existent Domain response in DGAs detection. Though NN approaches can achieve ideal accuracy, they require to collect and train massive data. That challenges defenders to make efforts in reducing the training time for response in time. Besides, it is difficult to tell the new AGDs by the previous models, as the data of entirely new families are not trained when DGAs work rapidly .

## VI. Conclusion

In this paper, we motivate the needs for better classifying vowels and consonants in domain names in the context of AGDs detection. To address this challenge, we propose DOLPHIN following the mature principles of phonics. DOLPHIN introduces a mapping between the vowel/consonant classification and the spelling of graphemes. Based on such patterns, DOLPHIN then extracts the linguistics features from domain names. We conduct various experiments to train those features on OSINT DGA feed and Alexa data with various classifiers. Results shown that when detecting AGDs, DOLPHIN can achieve 0.0265 higher mean accuracy than the state-of-the-art approach with RF classifiers and can also generalize to other classifiers with similar improvements.


## Acknowledgment

The authors would like to thank Bembenek Consulting for granting us access DGA. This research was partially supported by the National Natural Science Foundation of China (No. U19B2025, 62172323, and 62102001), the Open-end Fund Project of Science and Technology on Communication Networks Laboratory (No. HHX20641X004), and Natural Science Foundation of Anhui Higher Education Institution (No. KJ2020A0037). The corresponding author is Yazhe Tang.


## References


[1] G. Jacob, R. Hund, C. Kruegel, and T. Holz, "Jackstraws: Picking command and control connections from bot traffic." in *20th USENIX Security Symposium (USENIX Security 11)*, vol. 2011. San Francisco, CA, USA, 2011.

[2] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2014, pp. 1–21.

[3] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover," in *Proceedings of the 16th ACM conference on Computer and communications security*, 2009, pp. 635–647.

[4] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 48–61.

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

[6] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "Fanci: Feature-based automated nxdomain classification and intelligence," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1165–1181.

[7] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," *arXiv preprint arXiv:1611.00791*, 2016.

[8] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: Dga-based botnet tracking and intelligence," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2014, pp. 192–211.

[9] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: detecting the rise of dga-based malware," in *21th USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 491–506.

[10] P. R. Hanna *et al.*, *Phoneme-grapheme correspondences as cues to spelling improvement*. ERIC, 1966.

[11] R. S. Berndt, J. A. Reggia, and C. C. Mitchum, "Empirically derived probabilities for grapheme-to-phoneme correspondences in english," *Behavior Research Methods, Instruments, & Computers*, vol. 19, no. 1, pp. 1–9, 1987.

[12] M. Patricia, E. Witting, and L. Stehr, *Phonics they use: Words for reading and writing*. Pearson, 1995.

[13] E. Fry, "Phonics: A large phoneme-grapheme frequency count revised," *Journal of Literacy Research*, vol. 36, no. 1, pp. 85–98, 2004.

[14] BembenekConsulting, "Dga domain feed," [EB/OL], https://osint.bambenekconsulting.com/feeds/ July, 2019.

[15] F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "Botfinder: Finding bots in network traffic without deep packet inspection," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012, pp. 349–360.

[16] J. Kwon, J. Lee, H. Lee, and A. Perrig, "Psybog: A scalable botnet detection method for large-scale dns traffic," *Computer Networks*, vol. 97, pp. 48–73, 2016.

[17] B. Yu, D. L. Gray, J. Pan, M. De Cock, and A. C. Nascimento, "Inline dga detection with deep networks," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 683–692.