Design of Malicious Domain Detection Dataset for Network Security

Ju-Chien Cheng[1][0000−0001−7129−5836], Naoto Yanai[1][0000−0002−0817−6188], and Shingo Okamura[2]

[1] Osaka University, 1-5 Yamadaoka, Suita, Osaka, Japan.
yanai@ist.osaka-u.ac.jp
[2] National Institute of Technology, Nara College, 22 Yata-cho, Yamatokoriyama, Nara, Japan.

Abstract. The design of a domain dataset is an essential work for research on malicious domain detection. However, the design of a dataset is a heavy task in general, and the dataset itself is often non-disclosure because cybersecurity-related information is popularly sensitive. In this paper, we design a dataset that is publicly available for research on malicious domain detection. We also confirm the validity of the dataset through reference implementation of machine learning algorithms about the classification of malicious and benign domains. Furthermore, we discuss feature selection which affects the domain classification, and then show an insight into the dataset to improve the classification performance. In particular, we demonstrate that a model based on LightGBM can classify benign and malicious domains with 94.3% accuracy by collecting 100,000 benign data and 24,126 malicious data. Meanwhile, when 24,126 domains for each data are utilized, F1-score on the classification becomes 10% higher than the case where all the data are utilized. Moreover, when we reconstruct a classification model with only features whose importance is high, .e.g., the number of NS records, a similar accuracy can be provided despite removing the features by half.

Keywords: Malicious Domain Detection, Dataset, Feature Engineering, Machine Learning.

1 Introduction

1.1 Backgrounds

In proportion to cybercrime, malicious domains have increased rapidly in the past decade. For example, after 2020, there are many domains piggybacking on COVID-19[3]. Detection and countermeasures against these malicious domains are a serious and significant issue in modern cybersecurity.

[3] https://www.itproportal.com/news/thousands-of-malicious-covid-19-domains-hosted-on-public-clouds/

A typical countermeasure is to make a blacklist of malicious domains to block access to the domains. However, making a blacklist will be no longer a viable approach in many situations since novel domains continually appear through domain generation algorithms (DGAs). Based on the background above, machine learning is recognized as being the most attractive approach to malicious domain detection in recent years [7, 34, 38]. Meanwhile, a dataset for benign and malicious domains is crucial for research on domain detection in the field of machine learning.

Nevertheless, datasets are non-disclosure in most of early literature [38]. According to some survey [38], there are two scientific problems due to the non-discloses. First, current research cannot reproduce a tool and its experimental results and compare them with those in early literature. In general, scientific advances rely on the validation of and comparison with the existing approach. Despite this, most of the existing works significantly lack extensive and systematic experimental validation and comparison of different techniques due to the lack of the underlying dataset. Second, many researchers have no dataset required for domain research, and consequently, they cannot participate in the research itself. In general, domain data are continuously updated, and the number of domain themselves have increased rapidly. However, many researchers do not have resources for data collection, and thus conducting experiments is challenging already. In other words, data collection is a barrier against entry to domain research for many researchers. Based on the background described above, if there is a publicly available dataset, it will provide novel research and entry for new researchers.

Meanwhile, precise extraction and selection of features for a dataset are often non-trivial. Notably, this is especially challenging in the field of malicious domain detection [38]. Loosely speaking, dividing each domain data into benign and malicious is insufficient, and thus a dataset must contain meaningful features whereby domains generated by DGA can be distinguished precisely. More specifically, we should evaluate a dataset from the quantitative standpoint [10] of how many malicious domains are recognized precisely. However, a method to select features for the desired performance on domain detection has never been presented, to the best of our knowledge. Even if a dataset becomes publicly available, it may be unsuitable for research on domain detection as long as meaningful features are unknown. Thus, the design of a dataset should also take into account meaningful features for domain detection.

1.2 Contribution

In this paper, we design a dataset for malicious domain detection, which is publicly available. Furthermore, we discuss features concerning domain detection from importance through reference implementation of domain detection algorithms. Hence, our findings take a new look at features included in a dataset appearing in the future.

This paper makes two contributions. The first contribution is to design the dataset itself. In the design of the dataset, we have utilized only publicly available data sources and have not included sensitive information such as IP addresses and hostnames concerning cybersecurity ethics. As described in detail in Section 3, we have also collected data as carefully as possible from the standpoint of cybersecurity ethics. Our dataset is available for only researchers who aim to do subsequent works by telling the authors how the dataset is utilized. In other words, we do not release unlimitedly to prevent the dataset from being utilized for malicious usage. Besides, we release reference implementation of malicious domain detection. Researchers who want to try domain detection research are encouraged to refer to our implementation.

The second contribution is to obtain the following key insights with respect to a dataset of domain detection through discussion on features. Specifically, we found that features related to DNS records, primarily the number of name servers, strikingly affect domain detection accuracy. Meanwhile, for features related to characteristic features of domain names, only the length of domain names, the entropy [32], and the reputation value [37] provide high importance. Moreover, we partially consider the use-case and limitation of our dataset for subsequent works. (See Section 4 for detail.)

2 Problem Setting

In this section, we describe domain names and machine learning as background knowledge to understand this work.

2.1 Domain Data

A domain name is information operated by decoupling the physical location such as an IP address of a service and its logical address and is operated via the domain name system (DNS). In general, domain names are hierarchically managed under namespaces called a zone, and the highest domain is called root. The most popular domains are `.com`, `.us`, and `.jp`, and such domains are called top level domains (TLDs). There are plural domains under each TLD, and hence domains are managed hierarchically and distributively through their zones.

2.2 Domain Classification by Machine Learning

We focus on domain detection based on machine learning, whereby domains are detected as benign and malicious. Roughly speaking, a model learns given data as the training process and then is able to infer whether an unseen data is benign or malicious as the inference process. A model is often trained with domain data obtained from publicly available sources. The problem statement in this paper is formulated as follows:

Problem Formulation Let a set of domain data be $\mathcal{D}$, a set of features be $\mathcal{F}$, the size of $\mathcal{F}$ be $|\mathcal{F}|$, features for each domain $d_i \in \mathcal{D}$ be $F_i = \{f_1^i, \cdots, f_l^i\} \subseteq \mathcal{F}^l$ for any $l \in \mathbb{N}$, a label of benign data be L_i^b, and that of malicious data be

L^m_i. Then, a model M for domain detection is a function which, given a tuple $DF = \{(d_1, F_1, L_1), \cdots, (d_n, F_n, L_n)\}$ of domain, feature, and a label for any integer $n \in \mathbb{N}$, a domain $d_t \in \mathcal{D}$ to be classified, and its features $F_t \subseteq \mathcal{F}$ as input, outputs a set $\{\epsilon^i_{d_t}\}_{i \in \{b,m\}} \subseteq \mathbb{R}^2$ of real numbers, i.e., $M(DF, d_t, F_t) \rightarrow \{\epsilon^i_{d_t}\}$, where $L_j \in \{L^b_j, L^m_j\}$ for any $j \in [1, n]$ and $\epsilon^i_{d_t}$ is a probability about benign/malicious with respect to d_t

2.3 Key Question

The design of a dataset is described as constructing a publicly available set $\mathcal{D}$ on the above formulation. The problems described in Section 1 mean that $M(DF, d_t, F_t)$ in early literature cannot be reproduced in its subsequent works due to an unpublished dataset $\mathcal{D}$.

Then, the key question in this paper, i.e., feature selection, is to find a subset $F_i \subseteq \mathcal{F}$ of features to maximize $M(DF, d_t, F_t) \rightarrow \{\epsilon^i_{d_t}\}$ for each domain d_i. In doing so, the computation of $M(DF, d_t, F_t)$ often requires a heavier cost in proportion to the size of F_i. Therefore, we discuss F_i with a small space such that a fairly practical $M(DF, d_t, F_t) \rightarrow \{\epsilon^i_{d_t}\}$ is obtained, as well as maximizing $\epsilon^i_{d_t}$.

In addition to the above question, we also have to care about data sharing as the technical challenge for designing a domain dataset [38]. Generally speaking, cybersecurity-related information is sensitive data, and hence it should not be shared in a public way. For example, by releasing information about a vulnerability, a device or network which are sources of the vulnerability may be attacked. Namely, even if a researcher is able to gain access to domain data from an Internet service provider (ISP) and wants to provide reproducibility for other researchers, it would often be extremely difficult to share the data.

Based on the above technical background, the design of a public dataset was negative in early literature. Although currently there are several publicly available domain datasets [14, 17], they are no longer updated.

3 Design of Dataset

In this section, we describe the data collection and the features of our dataset. After that, we discuss the ethical consideration about the data collection that occurred in this paper.

3.1 Infrastructure

The data collection was performed based on domains we gathered from public domain lists. We define our dataset only to contain either benign or malicious domains, where benign domains are those listed in top sites lists, and malicious domains are those listed in blacklists. Specifically, we use Tranco [27] as our benign domain list. For malicious domain list, we merged the domains from three public lists, which are URLhaus[4], CYBERCRiME-TRACKER[5], and PhishTank[6]. For those domains found in both lists, we labeled them both benign and malicious. Below we give brief overviews of the lists we used in this paper.

Tranco. Tranco is a ranking site mainly providing data for research. The data Tranco provides are obtained from multiple providers: Alexa, Cisco Umbrella, Majestic, and Quantcast. Users can query the data provided by Tranco through their API. The data list is provided in CSV format. By registering, the URLs of CSVs will be sent to the registered mail address. A Python package is also provided, allowing users to write custom python codes to work with the Tranco list. At the time of writing (November 2020), Quantcast data is not available and not included in the Tranco list. Therefore, in this paper, we only used data based on Alexa, Cisco Umbrella, and Majestic.

URLhaus. URLhaus is a project collecting, tracking, and sharing URLs used for malware distribution. The data is available in CSV format through their public API, licensed under CC0. The data from URLhaus is also used by commercial services such as Google Safe Browsing[7].

CYBERCRiME-TRACKER. CYBERCRiME-TRACKER lists information about control and command servers, a kind of server used to manage botnets such as ZeuS. According to [9], CYBERCRiME-TRACKER is used by malware analyzing service Virus Total[8] since 2013. At the time of writing (November 2020), CYBERCRiME-TRACKER contains 22472 URLs, available in TXT format.

PhishTank. PhishTank is a community-operated by OpenDNS, collecting information about phishing attacks. Information is reported by users and verified by the community. The data is available in XML, Serialized PHP, CSV, JSON through their API.

3.2 Data Collection

We collected corresponding DNS data based on the benign and malicious lists we obtained in the previous section. Specifically, we collected IP addresses, MX server addresses, PTR records, nameservers, and TTL (time to live). Of the data we collected, we only recorded the amount of distinct data and characteristics of the data, excluding information that can uniquely identify their data source from our dataset, i.e., raw addresses and domain names.

Generally, there are two methods to collect DNS data, namely active DNS and passive DNS. Active DNS is a method that uses a domain list and directly queries DNS resolvers to gather DNS data. On the other hand, passive DNS is usually deployed in corporations or organizations, collecting and analyzing DNS logs produced by user querying DNS. Since data collected by using passive DNS

[4] `https://urlhaus.abuse.ch/`

[5] `https://cybercrime-tracker.net/`

[6] `https://www.phishtank.com/`

[7] `https://urlhaus.abuse.ch/about/`

[8] `https://www.virustotal.com/gui/`

Table 1. Text-based features

Feature name	Definition
Length of domain	The length of the domain
Vowels	The number of vowels, the number of vowel characters, and the ratio of vowel characters in the domain. For example, "yahoo.co.jp" would be 3, 2, 0.6.
Constants	The number of constants and the number of constant characters in the domain. For example, "yahoo.co.jp" would be 2, 2.
Conversions of vowels and constants	The number of vowel-constant and constant-vowel pairs in the domain. For example, "yahoo.co.jp" would be 3.
Numbers	The number and ratio of numeric characters in the domain.
Conversions of numbers and alphabets	The number of digit-alphabet and alphabet-digit pairs in the domain.
Number of other characters	Number of characters other that digits and alphabets in the domain.
Length of max consecutive character	Maximum length of identical consecutive characters.
Entropy	The entropy [32] of the domain, defined by equation (1).
Reputation Value	The value representation of the semantics of the domain [37], defined by equation (2).

are data produced by real internet usages, it is close to real-world scenarios and more comfortable to be used to generate time-series data than active DNS data. However, passive DNS data are extracted from user logs, which leads to serious privacy concerns, making passive DNS data difficult to safely used outside the organization where data is collected. Therefore, in this paper, we only use active DNS.

Features We classified the features we extracted into three categories: text-based features, DNS-based features, and web-based features. The definition of these categories of features are listed in table 1, 2, 3 respectively. In general, text-based features are extracted from the string of the domain name, DNS-based features are extracted from DNS records of the domain name, and web-based features are extracted from web contents related to the domain. Note that in text-based features, the TLD part of domains is removed.

In table 1, entropy is defined as below.

$$Entropy = -\sum_{j=1}^{n_i} p^i_j \times \log 2(p^i_j), \tag{1}$$

Here n_i stands for the number of distinct characters, p^i_j stands for the number of occurrences of c^i_j divided by the length of the domain, that is, p^i_j =

Table 2. DNS features

Feature name	Definition
Number of IP	The number of distinct IP addresses.
Number of MX	The number of distinct name servers.
Number of NS	The number of distinct MX servers.
Number of PTR	The number of distinct PTR records, queried using each of the distinct IP addresses.
NS similarity	The similarity between name servers.
Number of countries	The number of countries obtained from GeoLite2[9], queried using each of the distinct IP addresses.
Mean of TTL	The average number of TTL.
Standard deviation of TTL	The standard deviation of TTL.

Table 3. Web-based features

Feature name	Definition
Number of HTML elements	The number of HTML elements of the content, obtained by accessing the domain.
WHOIS life time	The difference of expiration date and creation date of WHOIS data [33], in days.
WHOIS active time	The difference of update date and creation date of WHOIS data [33], in days.

$count(c^i_j)/length(Domain)$. Reputation value is defined as below.

$$RV = \sum_{i=1}^{m} W_N(i), \quad W_N(i) = \log 2 \left(\frac{C_N(i)}{N} \right), \tag{2}$$

Here $C_N(i)$ stands for the frequency of substring of i-th domain in domain ranking sites such as Tranco. Also noted that "." is excluded from the domain used text-based features.

Feature Selection Below we describe the reason that we selected these features. First of all, when we analyzed the domains from the domain list we collected, we found that the average length of malicious domains is about two times that of benign domains. Moreover, the average number of numeric characters in malicious domains is about five times that of benign domains. According to these findings, we considered features extracted from the text information of domains to be useful for classifying benign domains and malicious domains. For instance, [6] utilized domain names to analyze malicious domains. We expect that in this paper, we would be able to see the effectiveness of this kind of feature.

On the other hand, when analyzing DNS data of the selected domains, we found that the number of distinct IP addresses, MX records, PTR records, and NS records tend to be higher for benign domains. Especially for MX records and NS records, the difference is significant. The average number of distinct

IP addresses of benign domains is also 1.5 times that of malicious domains. We consider this due to multinational corporations or large enterprises seeking to utilize their internet infrastructures fully. For the above reasons, we consider DNS information to be useful features in this paper. In practice, [36] also showed the high importance of DNS records.

Lastly, we describe the reason for choosing web-based features. We compared the HTML contents of benign domains and malicious domains, and we found that the number of HTML tags is significantly higher in benign domains. A similar insight is also discussed by Wang et al. in [36]. Meanwhile, lifetime and active time were discussed by Shi et al. in [33] although the importance of these two factors was not discussed in deep. When analyzing the lifetime and active time of domains, we found a notable difference between that of benign domains and that of malicious domains. Intuitively, malicious domains, which are often used for distributing malware, do not require rich HTML contents to function. The contents of malicious domains also do not need to be updated regularly. These might be the reason for the findings above.

Collected Data The domain list used in this paper was collected on November 11, 2020. The list contains 100,000 benign domains and 24,126 malicious domains. Benign domains are obtained from Tranco's daily list (top 1M), and we further extracted the top 100,000 domains. Malicious domains are collected from URLhaus, PhishTank, and CYBERCRiME-TRACKER, where we excluded those without domain names. There were 4,872 domains from URLhaus, 4,622 domains from PhishTank, and 18,969 domains from CYBERCRiME-TRACKER. After removing these domains, the number of malicious domains became 24,126. Note that we did not exclude domains presenting in both the benign domain list and malicious domain list. Rather than that, this kind of domain is labeled both benign and malicious. Feature extraction was also performed on the same day. We use public DNS services such as 1.1.1.1 (provided by Cloudflare and APNIC), Google Public DNS, and OpenDNS to collect DNS features. For the extraction of countries of IP, we used GeoLite2 data provided by MaxMind[10].

3.3 Ethical Consideration

We discuss the ethical considerations when constructing our dataset below.

First of all, even though the Tranco list is published as a cybersecurity research-oriented list, it is based on commercial purpose services such as Alexa. In this paper, we used Tranco as a benign domain dataset but took the approach to label the domains as both benign and malicious as long as they are also listed in the malicious domain lists. Given that services such as Alexa are for commercial purposes, and in this paper, we are labeling some data from those services as malicious, our dataset may potentially damage the effectiveness of those products. However, we may also bring merits to the providers. Specifically, by analyzing malicious-labeled domain further, the providers may be able

[10] http://www.maxmind.com

to find potential malicious service undetected previously. Hence, the above approach may also be beneficial for improving the ranking of related services.

Meanwhile, we also need to consider the circumstance that one uses our dataset and wrongly determines benign domains as malicious domains. To deal with this kind of issue, we decided only to distribute our dataset upon request by researchers. After receiving requests, we then determine the usage of the dataset would be purely for research purposes before we send out our dataset. As a consequence of that, we eliminate the potential risks of our dataset being misused by the public. Also, the features used in this paper, as described in 3.2, are selected since they are likely to be general characteristics of malicious domains. From this perspective, we would like to give feedback to the owners or organizations whose domains are wrongly classified as malicious domains to reconsider their configurations. We also recommend researchers willing to use our dataset to do so. Like we described in the last paragraph, we urge to help society to find potential malicious services. We could prevent benign domains wrongly classified as malicious domains in the future by suggesting a reconsideration of configurations of domains.

Also, in our dataset, we did not include IP addresses or hostnames to prevent direct links to specific organizations, thereby ensuring that our dataset would not reveal specific domains. By looking inside our dataset, one can not simply trace back to the original domain, ensuring anonymity to a certain extent. Moreover, each data in our dataset, like described previously, are queried and collected by the authors. We did not collect data or logs from organizations or directly from the public. Hence, during our data collection, we can ensure that we are not invading others' privacy.

Lastly, the dataset constructed in this paper will not be directly open to the public but will be open to researchers interested in performing research about domains. We will ask the research purpose and ensure there is a proper purpose before sharing our dataset. As such, we prevent others from using our dataset for improper uses.

4 Experiments

In this section, as a case study based on the dataset constructed in the previous section, we implemented and evaluated a malicious domain detection algorithm using machine learning.

4.1 Experimental Purpose

The purpose of this experiment is to evaluate the effectiveness of the dataset in domain detection. Also, we would like to determine the effective features in classification.

To achieve that, first, we build a domain classification algorithm using machine learning, then we evaluate the model using different sizes of data and show the tendency. Specifically, we use LightGBM [15] to implement the algorithm.

Table 4. Experiment Results

	Accuracy	**Precision**	**Recall**	**F1**
(1)	94.26%	89.70%	79.58%	84.34%
(2)	93.66%	94.42%	92.83%	93.62%
(3)	91.03%	92.32%	89.45%	90.86%

After that, we evaluate the importance of individual features. Generally speaking, feature importance could be affected by the data of use and the parameters of the algorithm. Therefore the evaluation is for the algorithm in the previous paragraph. Lastly, we use features with high importance and perform classification again, comparing the result to the original result. If these results do not have a meaningful difference, then we show that we could use those features with high importance.

4.2 Effect of Number of Data

Our dataset is highly imbalanced, containing 100,000 benign domains, but only 24,126 malicious domains. Considering the imbalance, the experiments conducted in this paper will include all these three patterns: (1) using all of the benign domain data, (2) using top 24126 of benign domain data, and (3) using randomly selected 24126 of benign domain data.

We performed experiment using LightGBM [15] classifier and the three patterns above as input. We also applied 10-fold cross-validation in order to produce stable results. The results are shown in table 4.

Among all three patterns, (2) using the top 24126 of benign domain data results in the best numbers. We consider this to be that, as our perspective selecting features in 3.2, domains that exist at the top of Tranco may be showing a strong tendency to be "benign." They have more IP, MX, NS records, or have a higher WHOIS lifetime than those in lower rankings of Tranco. On the other hand, despite benign data in (1) and (3) should have similar distributions, the results are quite different. This shows that when performing domain classification, the amount of data is not the only determining factor. The balance of data also plays an important role here.

4.3 Effect of Feature Selection

The feature importance extracted from the LightGBM classifier in the previous section is shown in 5. This importance is obtained directly from API provided by LightGBM. Note that we applied L2 normalization to the values and rounded to the fourth decimal place in order to compare between patterns. The patterns are the same as in the previous section.

According to Table 5, we can see that the number of NS, the reputation value, and the mean of TTL have high importance. Especially for the number of NS, the importance is much higher than other features. Also, despite that the

Table 5. Feature Importance

Feature	(1)	(2)	(3)
Length of domain	0.0721	0.1034	0.0758
Number of vowel characters	0.0126	0.0106	0.0172
Number of vowels	0.0094	0.0074	0.0115
Ratio of vowel characters	0.0378	0.0289	0.0485
Number of constant characters	0.0177	0.0154	0.0232
Number of constants	0.0130	0.0117	0.0179
Conversions of vowels and constants	0.0476	0.0354	0.0621
Number of numeric characters	0.0159	0.0042	0.0142
Ratio of numeric characters	0.0044	0.0041	0.0055
Conversions of numbers and alphabets	0.0120	0.0074	0.0130
Number of other characters	0.0065	0.0044	0.0064
Length of max consecutive character	0.0092	0.0054	0.0097
Entropy	0.0663	0.0612	0.0811
Reputation Value	0.2274	0.1108	0.2451
Number of IP	0.0323	0.0379	0.0434
Number of MX	0.0436	0.0612	0.0539
Number of PTR	0.0361	0.0505	0.0397
Number of NS	0.9020	0.8913	0.8604
NS similarity	0.0619	0.0954	0.0850
Number of countries	0.1731	0.0864	0.2024
Mean of TTL	0.1696	0.2182	0.2019
Standard deviation of TTL	0.0949	0.0940	0.0973
Number of HTML elements	0.1328	0.1413	0.1770
WHOIS life time	0.1467	0.2688	0.1894
WHOIS active time	0.0723	0.0569	0.0857

tendencies of feature importance do not differ much between (1) and (3), (2) has a different distribution. We consider the reason for this is that, as our perspective selecting features in Section 3.2, rather than the whole list, domains at the top of the list showed a stronger tendency. This may also be the reason that in (2), the importance of the reputation value and the number of countries became lower, and that of Length of domain and WHOIS lifetime became higher.

Next, we re-trained all of the models using features with important values, as shown in 5, higher or equal to 0.05. As a result, all of the DNS features and web-based features are selected. However, for text-based features, (1) and (3) would be only using the length of domain, the number of numeric characters, the entropy, and the reputation value, while (2) would be only using the length of domain, the entropy, and the reputation value. The result is shown in Table 6. According to the result, especially for the column of (2), the values only changed less than 0.5% compared to 4. We consider that to be the same level of effectiveness. This means that most of the text-based features, which we did not select above, made only little contributions to the classification algorithm's results. Therefore, when constructing the dataset, we could exclude this

Table 6. Experiment Results

	Accuracy	Precision	Recall	F1
(1)	93.88%	89.26%	77.99%	83.21%
(2)	93.33%	94.22%	92.42%	93.29%
(3)	90.55%	92.07%	88.74%	90.35%

information. That is, from the perspective of effectively constructing a dataset, reducing the amount of text information obtained would also be an important factor.

4.4 Use Case and Limitation

Our dataset is designed for use on the client-side. For example, we can design a browser module to detect connections to malicious domains. We are currently working on an application to perform detection in a browser and have released its prototype via GitHub[11]. We plan to keep developing this application in the future. There would also be other use cases, including a firewall system that prevents clients from connecting malicious domains from being deployed by network administrators on network gateways.

There are two major limitations of this paper. First, we did not consider the behavior of the clients before or after connecting malicious domains. Therefore, the dataset in this paper would not be applicable for applications based on client behaviors such as [22, 25]. Next, we only utilized active DNS. Analyses that look at the relations of active DNS data and passive DNS data [16] cannot be performed using our dataset. To perform such an analysis, a passive DNS dataset is required in addition to an active DNS dataset.

5 Related Work

5.1 Domain Detection

Many of the researches of malicious domains utilize information obtained from DNS services, certificates, the structure of web pages, and external sources [38]. Although DNS information, as described in section 3, is often used, for malicious sites, there is known to be specific tendency of the behaviors of communication between clients and DNS servers [23, 26, 28, 29]. There are also works transforming these information to formats that are easier to handle, such as determinant expression or graphs. For example, in [11], malware detection algorithm that utilizes knowledge base DGA (Domain Generation Algorithm) was proposed. To perform detection, [11] used an analytical approach, that the infected hosts can be distinguished from normal hosts by using the balance of the number of IP addresses and DNS queries. In [6], a prediction model using random forest based on the time series of domains recorded on trusted lists, was proposed.

[11] `https://github.com/kzk-IS/MADMAX`

There are also researches regarding malicious domain detection which used certificate information. For example, [35] built a classification model using features extracted from TLS communication, which includes certificates, and http communication. Also, [18] detects phishing sites using only certificate information. There also exist researches of detection of malicious sites such as phishing sites using the structure of web pages. For instance, [13] compared the text, fonts, and other information from web pages to trustworthy sites. [24], on the other hand, aimed to detect malicious sites by making comparisons of CSS (Cascading Style Sheet) between malicious and benign sites. However, these approaches did not work well against obfuscation of codes [8, 20].

On the other hand, researches attempted to perform detection using graphical information of web pages also exist. For example, in [2], SIFT (Scale Invariant Feature Transform) was applied to extract features for malicious site detection, from the logos of web pages. Research that applied approaches such as SURF, Histogram of Oriented Gradients, and Contrast Context Histogram to screenshots to extract features is also proposed to classify domains [30, 5, 4, 21]. However, it is predicted that in the field of image processing, learning using CNN is going to yield higher precision than these features extracted by human work [19, 31]. Therefore, in [1], based on image information of benign sites and malicious sites, a model is built using CNN to detect phishing sites.

5.2 Design of Dataset

As publicly available dataset about DNS, [17] and [14] were proposed. [17] utilized active DNS, while [14] utilized passive DNS. These researches share similar motivation as ours, urging to design dataset aimed for researches related to domains such as domain classification. In Japan, starting by the community of Anti Malware Engineering WorkShop (MWS), dataset designed for researches of malware are proposed [12]. Despite that the dataset is still getting updates now in 2020, D3M, which contains domain information, has not being updated since 2015.

6 Conclusion

In this paper, we designed a publicly available dataset for domain detection. We also provided a reference implementation of domain detection algorithms as a case study. Consequently, for researchers who aim domain research, we can provide a dataset design, which is a barrier against entry to domain research, and the reproducibility of results in early literature. We plan to freely share our dataset with researchers in accordance with their contacts and research motivation.

Besides, we obtained several insights via designing the dataset. First, for DNS records features, the number of name servers affects the accuracy of domain detection strikingly. Second, only the length of domains, the entropy, and the reputation values provide high scores for the importance of strings as domain names. We believe that these insights will support the further design of a dataset and subsequent malicious domain detection works.

We are currently investigating features for domain detection in more detail. In particular, we will shed light on how other features except for the length, the entropy, and the reputation value affect detection results through further experiments. Further studies, which take the importance of features from various aspects into account, will need to be undertaken. More specifically, we plan to more find the importance of features through evaluation of permutation importance [3].

Acknowledgement This work was supported in part by the Japan Society for the Promotion of Science KAKENHI Number 18K18049, and the Innovation Platform for Society 5.0 at MEXT.

References

1. Abdelnabi, S., Krombholz, K., Fritz, M.: Visualphishnet: Zero-day phishing website detection by visual similarity. In: Proc. of CCS 2020. p. 1681–1698. ACM (2020)
2. Afroz, S., Greenstadt, R.: Phishzoo: Detecting phishing websites by looking at them. In: Proc. of ICSC 2011. pp. 368–375. IEEE (2011)
3. Altmann, A., Toloşi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. Bioinformatics **26**(10), 1340–1347 (2010)
4. Bozkir, A.S., Sezer, E.A.: Use of hog descriptors in phishing detection. In: Proc. of ISDFS 2016. pp. 148–153. IEEE (2016)
5. Chen, K.T., Chen, J.Y., Huang, C.R., Chen, C.S.: Fighting phishing with discriminative keypoint features. IEEE Internet Computing **13**(3), 56–63 (2009)
6. Chiba, D., Yagi, T., Akiyama, M., Shibahara, T., Yada, T., Mori, T., Goto, S.: Domainprofiler: Discovering domain names abused in future. In: Proc. of DSN 2016. pp. 491–502. IEEE (2016)
7. Curtin, R.R., Gardner, A.B., Grzonkowski, S., Kleymenov, A., Mosquera, A.: Detecting dga domains with recurrent neural networks and side information. In: Proc. of ARES 2019. ACM (2019)
8. Fu, A.Y., Wenyin, L., Deng, X.: Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). IEEE Transactions on Dependable and Secure Computing **3**(4), 301–311 (2006)
9. Gañán, C., Cetin, O., van Eeten, M.: An empirical analysis of zeus c&c lifetime. In: Proc. of AsiaCCS 2015. pp. 97–108. ACM (2015)
10. Grill, M., Nikolaev, I., Valeros, V., Rehak, M.: Detecting dga malware using netflow. In: Proc. of IM 2015. pp. 1304–1309. IEEE (2015)
11. Grill, M., Nikolaev, I., Valeros, V., Rehak, M.: Detecting dga malware using netflow. In: Proc. o IM 2015. pp. 1304–1309. IEEE (2015)
12. Hatada, M., Akiyama, M., Matsuki, T., Kasama, T.: Empowering anti-malware research in japan by sharing the mws datasets. Journal of Information Processing **23**(5), 579–588 (2015)
13. Huang, C.Y., Ma, S.P., Yeh, W.L., Lin, C.Y., Liu, C.T.: Mitigate web phishing using site signatures. In: Proc. of TENCON 2010. pp. 803–808. IEEE (2010)
14. Inc., F.S.: Dns database (2015), `https://wwww.dnsdb.info/`

15. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: Proc. of NIPS 2017. vol. 30, pp. 3146–3154. Curran Associates, Inc. (2017)
16. Khalil, I., Yu, T., Guan, B.: Discovering malicious domains through passive dns data graph analysis. In: Proc. of AsiaCCS 2016. p. 663–674. ACM (2016)
17. Kountouras, A., Kintis, P., Lever, C., Chen, Y., Nadji, Y., Dagon, D., Antonakakis, M., Joffe, R.: Enabling network security through active dns datasets. In: Proc. of RAID 2016. pp. 188–208. Springer (2016)
18. Kovar, D.H.R.: The " hidden empires " of malware (2018), `https://www.slideshare.net/RyanKovar/the-hidden-empires-of-malware-with-tls-certified-hypotheses-and-machine-learning`
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. of NIPS 2012. vol. 1, p. 1097–1105. Curran Associates Inc. (2012)
20. Lam, I.F., Xiao, W.C., Wang, S.C., Chen, K.T.: Counteracting phishing page polymorphism: An image layout analysis approach. In: Proc. of ISA 2009. LNCS, vol. 5576, pp. 270–279. Springer (2009)
21. Malisa, L., Kostiainen, K., Capkun, S.: Detecting mobile application spoofing attacks by leveraging user visual similarity perception. In: Proc. of CODASPY 2017. p. 289–300. ACM (2017)
22. Manadhata, P.K., Yadav, S., Rao, P., Horne, W.: Detecting malicious domains via graph inference. In: Proc. of ESORICS 2014. LNCS, vol. 8712, pp. 1–18. Springer (2014)
23. Manadhata, P.K., Yadav, S., Rao, P., Horne, W.: Detecting malicious domains via graph inference. In: Proc. of ESORICS 2014. LNCS, vol. 8872, pp. 1–18. Springer (2014)
24. Mao, J., Tian, W., Li, P., Wei, T., Liang, Z.: Phishing-alarm: Robust and efficient phishing detection via page component similarity. IEEE Access **5**, 17020–17030 (2017)
25. Oprea, A., Li, Z., Yen, T.F., Chin, S.H., Alrwais, S.: Detection of early-stage enterprise infection by mining large-scale log data. In: Proc. of DSN 2015. pp. 45–56. IEEE (2015)
26. Oprea, A., Li, Z., Yen, T.F., Chin, S.H., Alrwais, S.: Detection of early-stage enterprise infection by mining large-scale log data. In: Proc. of DSN 2015. pp. 45–56. IEEE (2015)
27. Pochat, V.L., van Goethem, T., Tajalizadehkhoob, S., Korczynski, M., Joosen, W.: Tranco: A research-oriented top sites ranking hardened against manipulation. In: Proc. of NDSS 2019. Internet Society (2019)
28. Rahbarinia, B., Perdisci, R., Antonakakis, M.: Segugio: Efficient behavior-based tracking of malware-control domains in large isp networks. In: Proc. of DSN 2015. p. 403–414. IEEE (2015)
29. Rahbarinia, B., Perdisci, R., Antonakakis, M.: Efficient and accurate behavior-based tracking of malware-control domains in large isp networks. ACM Transactions on Privacy and Security **19**(2) (2016)
30. Rao, R.S., Ali, S.T.: A computer vision technique to detect phishing attacks. In: Proc. of CSNT 2015. pp. 596–601. IEEE (2015)
31. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: An astounding baseline for recognition. In: Proc. of CVPRW 2014. pp. 512–519. IEEE (2014)
32. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review **5**(1), 3–55 (2001)

33. Shi, Y., Chen, G., Li, J.: Malicious domain name detection based on extreme machine learning. Neural Processing Letters **48**(3), 1347–1357 (2018)
34. Sun, X., Yang, J., Wang, Z., Liu, H.: Hgdom: Heterogeneous graph convolutional networks for malicious domain detection. In: Proc. of NOMS 2020. pp. 1–9. IEEE (2020)
35. Torroledo, I., Camacho, L.D., Bahnsen, A.C.: Hunting malicious tls certificates with deep neural networks. In: Proc. of AISec 2018. p. 64–73. ACM (2018)
36. Wang, Q., Li, L., Jiang, B., Lu, Z., Liu, J., Jian, S.: Malicious domain detection based on k-means and smote. In: Proc. of ICCS 2020. LNCS, vol. 12138, pp. 468–481. Springer (2020)
37. Zhao, H., Chang, Z., Bao, G., Zeng, X.: Malicious domain names detection algorithm based on *N*-gram. Journal of Computer Networks and Communications **2019**, 4612474:1–4612474:9 (2019)
38. Zhauniarovich, Y., Khalil, I., Yu, T., Dacier, M.: A survey on malicious domains detection through dns data analysis. ACM Computing Surveys **51**(4) (2018)