Received 6 October 2020; revised 7 November 2020; accepted 13 November 2020. Date of publication 17 November 2020; date of current version 9 December 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.3038704

A Superficial Analysis Approach for Identifying Malicious Domain Names Generated by DGA Malware

AKIHIRO SATOH [1], YUTAKA FUKUDA [1] (Member, IEEE), TOYOHIRO HAYASHI[1], AND GEN KITAGATA[2]

[1]Information Science and Technology Center, Kyushu Institute of Technology, Kitakyushu 804-8550, Japan

[2]Research Institute of Electrical Communication, Tohoku University, Sendai 980-8577, Japan

CORRESPONDING AUTHOR: A. SATOH (e-mail: satoh@isc.kyutech.ac.jp)

This work was supported by JSPS KAKENHI under Grant JP18K11296. Part of this work was carried out under the Cooperative Research Project of the RIEC, Tohoku University.

ABSTRACT Some of the most serious security threats facing computer networks involve malware. To prevent malware-related damage, administrators must swiftly identify and remove the infected machines that may reside in their networks. However, many malware families have domain generation algorithms (DGAs) to avoid detection. A DGA is a technique in which the domain name is changed frequently to hide the callback communication from the infected machine to the command-and-control server. In this article, we propose an approach for estimating the randomness of domain names by superficially analyzing their character strings. This approach is based on the following observations: human-generated benign domain names tend to reflect the intent of their domain registrants, such as an organization, product, or content. In contrast, dynamically generated malicious domain names consist of meaningless character strings because conflicts with already registered domain names must be avoided; hence, there are discernible differences in the strings of dynamically generated and human-generated domain names. Notably, our approach does not require any prior knowledge about DGAs. Our evaluation indicates that the proposed approach is capable of achieving recall and precision as high as 0.9960 and 0.9029, respectively, when used with labeled datasets. Additionally, this approach has proven to be highly effective for datasets collected via a campus network. Thus, these results suggest that malware-infected machines can be swiftly identified and removed from networks using DNS queries for detected malicious domains as triggers.

INDEX TERMS Domain generation algorithm, domain name system, malware, network security.

I. INTRODUCTION

SOME of the most serious security threats facing computer networks involve malware. Most commonly, cybercriminals control malware-infected machines through a command-and-control server (C&C) and use these machines to undertake malicious activities such as stealing confidential information, spreading malware to additional machines, and phishing within an organization. According to a recent McAfee report [1], over 300,000 new forms of malware and their variants are created each day, and the global annual cost of malware-related cybercrime may be as much as $600 billion. Thus, the establishment of security mechanisms to protect networks against malware encroachments is imperative.

To prevent malware-related damage, administrators need to swiftly identify and remove the infected machines that may reside in their networks. However, many malware families have domain generation algorithms (DGAs) to avoid detection [2]. A DGA is a technique in which the domain name is changed frequently to hide the callback communication from the infected machine to the C&C. Specifically, DGA malware dynamically generates and then attempts to

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

resolve domain names. The domain name that returns the correct response is considered the C&C. Since the lifetimes of the domain names generated for callbacks are extremely short, it is difficult for conventional security appliances that monitor communication with known malicious domains to detect callbacks caused by DGA malware.

Previously, deep packet inspection (DPI), which surveys packet payloads, has been used to strengthen protections against malware encroachments [3], [4]. However, in a cybersecurity report [5], Cisco noted that over 50% of Internet communications are encrypted and that approximately 70% of all malware programs encrypt their communications. Thus, the share of future communications amenable to DPI analysis will be reduced to a negligible subset, in inverse proportion to the fraction of encrypted communications.

In this article, we aim to identify dynamically generated domains from massive domain name system (DNS) queries to detect the callbacks of DGA malware. We focus on queried domain names for the DNSs because name resolution is an unencrypted interaction that always occurs prior to malware communication. Since the malicious domains generated by DGA malware have extremely short lifetimes, the main problem is the limitations of the available features for identifying these domains. Therefore, we propose an approach for estimating the randomness of domain names by superficially analyzing their character strings. Note that natural language processing techniques cannot be directly applied to the superficial analysis of domain names because domain names are short, unstructured, and unsegmented character strings. This approach is based on the following observations: human-generated benign domain names tend to reflect the intent of their domain registrants, such as an organization, product, or content. In contrast, dynamically generated malicious domain names consist of meaningless character strings because conflicts with already registered domain names must be avoided; hence, there are discernible differences in the strings of dynamically generated and human-generated domain names. Actually, the major DGA malware such as Emotet, Mirai, and Ramnit that caused a pandemic crisis has this characteristic in domain names [6], [7]. Notably, some previous studies have used a similar approach to discern distinctions between benign and malicious domains by considering the differences in character strings [8]. These studies employ machine learning (ML) or deep learning (DL) algorithms with training datasets to examine the nature of domain names.

Typical ML- and DL-based methods assume that the training dataset will exhibit the same characteristics as the actual dataset. However, substantial differences between training and actual datasets often arise for a variety of reasons, including the impossibility of preparing exhaustive datasets covering all events, the presence of strong biases in datasets observed as functions of time or space, and temporal variations in the trends exhibited by observed datasets. This phenomenon of divergence between datasets, which is known as concept drift [9], may prevent ML- and DL-based detection methods from achieving their full theoretical performance. Moreover, the task of aggregating correctly labeled datasets is unwieldy and cumbersome in practice [10]. In contrast, our approach does not require prior dataset training to distinguish between benign and malicious domains and thus prevents the possible impact of concept drift.

Our evaluation indicates that the proposed approach is capable of achieving recall and precisions as high as 0.9960 and 0.9029, respectively, when used with labeled datasets. Additionally, this approach has proven to be highly effective for datasets collected via campus networks. Taken together, these results suggest that malware-infected machines can be swiftly identified and removed from networks using DNS queries for detected malicious domains as triggers. Our approach contributes to dramatically improving network security by providing a technique to address various malware encroachment. The motivation, novelty, and contribution of this study can be summarized as follows.

- *Motivation:* We aim to identify dynamically generated domains from massive DNS queries to detect the callbacks of DGA malware. However, ML- and DL-based detection methods have been prevented from exhibiting their full potential performance levels because of the following problems: the cumbersome task of assigning labels to datasets and the phenomenon of concept drift, i.e., the discrepancies between training and actual datasets.
- *Novelty:* We propose a superficial analysis approach for identifying malicious domain names generated by DGA malware. A key point to emphasize is that our approach requires no prior information whatsoever regarding DGAs. Instead, it distinguishes between benign and malicious domains by determining whether or not a character string is meaningful. This dramatically mitigates the dataset-related problems that degrade detection performance levels.
- *Contribution:* In the experiments conducted as part of this study, our approach achieved recall and precision as high as 0.9960 and 0.9029 when used with labeled datasets. We also demonstrated its effectiveness in practical network operation. Taken together, these results suggest that malware-infected machines can be swiftly identified and removed from networks.

The remainder of this article is organized as follows. In Section II, we review the related studies and discuss their limitations. In Section III, to identify DGA-generated malicious domains, we propose an approach for estimating the randomness of domain names by superficially analyzing their character strings. Then, Section IV describes the experiments conducted to analyze the effectiveness of our proposed approach for detecting the callbacks of DGA malware. Finally, we summarize our conclusions and future work in Section V.

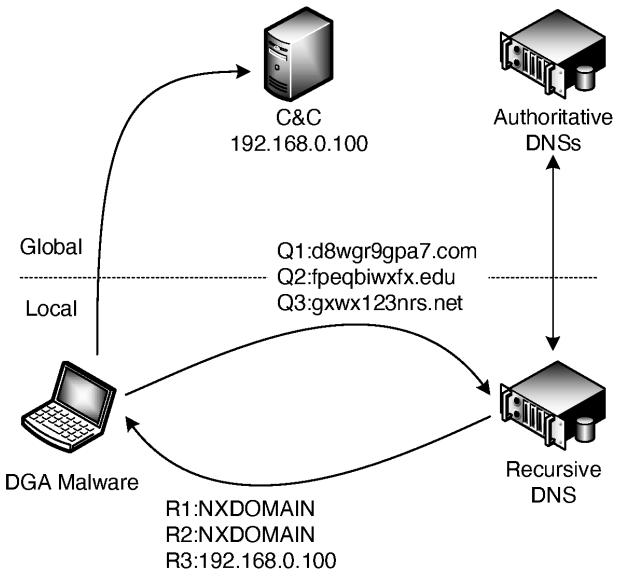

FIGURE 1. Callback communication between DGA malware and C&C.

II. BACKGROUND

Computer networks are vulnerable to malware hazards due to their interconnectivity and reachability. Thus, the research community has expressed strong ongoing interest to develop adequate defense solutions. We begin this section by describing the details of DGA malware; then, we introduce studies on detecting various malware and note their limitations.

A. DGA MALWARE

Malware programs such as Conficker and Kraken, which have caused considerable worldwide damage, implement DGAs as one aspect of their capabilities. Moreover, researchers have found improper codes embedded in websites and Web-based advertisements that are designed to promote the spread of such malware programs over broad target areas [11], [12].

Figure 1 schematically depicts callbacks implemented by DGA malware. In this figure, communications labeled **Q** are requests for name resolution from a DGA malware to a recursive DNS server (RDNS), while communications labeled **R** are responses to these requests. Here, we assume that DGA-generated domain names, being the capability of a C&C, are preregistered in an authoritative DNS server (ADNS). First, the malware dynamically generates multiple domain names, such as `d8wgr9gpa7.com`, `fpeqbiwxfx.edu`, and `gxwx123nrs.net`, based on its DGA and directs DNS queries regarding these domain names to the RDNS in the network to which the malware itself belongs. Upon receiving a query, the RDNS returns the address assigned to the domain name if it is registered in the ADNSs or a nonexisting domain (NXDOMAIN) response if it is not. Finally, any domain, e.g., `gxwx123nrs.net` in this figure, that corresponds to an affirmative DNS response is assumed to be the C&C, and the malware attempts callbacks for the address assigned to that domain.

The objective of the DGA is to establish highly available communication channels between the malware and the C&C. One result of this process is that communication barriers based on blacklists are easily avoided by changing C&C domains. Furthermore, communications from within the network to the outside span a wide range of destination addresses and are thus difficult to detect such that these communications are not restricted by firewalls. A key point note is that by using identical DGAs, the malware and the C&C do not need to exchange any information whatsoever to change the domain name.

B. RELATED WORK

Studies that focus on producing more sophisticated blacklists are frequently conducted and constitute the core of network threat defense strategies. For example, Soldo *et al.* [13] proposed a method for significantly enhancing the performance of blacklists based on previous attack logs provided by multiple contributors. Meanwhile, Freudiger *et al.* [14] improved the confidentiality of this method by sharing attack logs through peer-to-peer communications. Špaček *et al.* [15], [16] developed a DNS firewall system that blocks communications from a protected network to malicious domains on an outside network. This system uses DNS response policy zone (RPZ) technology [17] for advanced domain blacklisting. Unfortunately, malware families with DGAs are capable of avoiding blacklist-based detection by frequently changing the domain of their C&C.

In another study, Gu *et al.* [18] implemented BotHunter, a DPI-based passive network monitoring system that models typical malware behavior and interprets communications exhibiting strong associations with such behavior as evidence of malware contamination. Other reports have discussed efforts to improve the performance of DPI-based detection [19], [20]. However, the fraction of all communications amenable to DPI analysis has dwindled to negligible levels in inverse proportion to the role played by encryption on the Internet. This scenario has motivated a focus on the resolution of domain names, which are not affected by encryption, as a source of information for detecting malware.

Bilge *et al.* [21] developed a domain reputation system named Exposure that passively measures features from DNS traffic. Examples of such features include the number of addresses assigned to a domain, the lifetime of a domain, and the length of a domain name. This system uses a supervised learning algorithm, and its accuracy generally depends on the quantity and quality of training datasets. However, the malicious domains of DGA malware commonly return NXDOMAIN responses and have extremely short lifetimes, which make it difficult to secure a sufficiently robust training dataset.

Rahbarinia *et al.* [22] developed a system called Segugio that finds unknown malicious domains based on their co-occurrence relation with known malicious domains in

DNS queries. Segugio is based on the following insights: (1) infected machines in the same malware family tend to communicate with the same malicious domain group and (2) uninfected machines have no reason to communicate with malicious domains. However, for DGA malware, the temporary malicious domains used for callbacks have extremely short lifetimes, and the domains that co-occur with temporary malicious domains may not actually exist. Consequently, the system is not sufficient to detect the callbacks of DGA malware.

Berger *et al.* [23] developed a system called DNSMap that discovers potentially compromised machines based on DNS traffic. DNSMap identifies suspicious agile DNS mappings, i.e., mappings characterized by rapidly changing domain names and/or addresses, which are often used by the C&C. Meanwhile, Wang *et al.* [24] deployed a system called DBod, which classifies and detects DGA malware based on the analytical results of DNS query behavior. Specifically, since machines contaminated by the same malware family tend to generate a large number of identical DNS queries by the same DGA, their queries also tend to exhibit a similar domain scope and distribution. Thus, DBod exploits these similarities for classification and detection. However, since the two systems require the observation of an enormous amount of extensive DNS traffic, their operation is limited to large-scale networks such as Internet service providers (ISPs).

Plohmann *et al.* [25] revealed the DGA landscape by reverse-engineering 43 malware families, and Zago *et al.* [26] provided a mature dataset with analysis results for over 30 million domains generated by 50 malware families. Based on those results, other previous studies have attempted to distinguish between benign and malicious domains using only their character strings in manners similar to our approach. Truong and Cheng [27] proposed a technique that learns and predicts character patterns using bigram models with supervised learning algorithms, and Anderson *et al.* [28] extended this technique using character-level models with long short-term memory (LSTM) networks. Qiao *et al.* [29] combined LSTM networks with attention mechanisms to give proper weight values to the characters in domain names. Li *et al.* [30] implemented a framework for classification, detection, and prediction by combining multiple ML and DL algorithms. In addition, Vinayakumar *et al.* [8] compared the performance of various ML and DL algorithms in detecting DGA-generated malicious domains. These techniques are based on the existence of discernible bias in the rules for generating malicious domain names and thus need to learn the bias in advance by analyzing both benign and malicious datasets.

Such ML- and DL-based detection methods have two disadvantages. First, the task of aggregating labeled datasets is unwieldy and cumbersome in practice [10]. Based on an analysis of the latest studies, Sivaguru *et al.* [31] raised doubts about the practical utility of DGA classifiers, which use whitelists and blacklists as datasets. Their study concluded that because whitelisted domains are not sufficiently representative of the benign domains observed in actual networks, datasets in which labels are assigned by analyses, such as in the work of Yu *et al.* [32], yield results that are superior to those obtained using whitelists. However, because benign domains are specific to particular networks, the task of labeling them requires a deep audit and analysis of communications in those networks, which is an extremely time-consuming process in itself. A second drawback is the phenomenon of concept drift [9] between datasets, i.e., discrepancies between training datasets and actual target datasets. Typical ML- and DL-based methods assume that the training datasets exhibit the same characteristics as the actual datasets. However, substantial differences between training and actual datasets arise in many cases for a variety of reasons, including the impossibility of preparing exhaustive datasets covering all events, the presence of strong biases in datasets observed as functions of time or space, and temporal variation in the trends exhibited by observed datasets. These differences may prevent ML- and DL-based detection methods from achieving their full theoretical performance levels. Learning under concept drift is one field of growing interest [33], [34]. In particular, concept drift frequently arises in security-related datasets due to changes in malware activities. For example, Pendlebury *et al.* [35] considered the classification of Android malware and reported that the expected accuracy levels were difficult to attain in real-world environments.

TABLE 1. Examples of domain names generated by DGA malware.

ChinAd	7qvdqaw561dtasyi.com
	xe0d7fazyrvvwl9f.ru
Conficker	pfnnwjoeuee.biz
	xjjjvqpoh.com.ai
Locky	jlbroeji.biz
	btlwubflhfllshn.info
NewGOZ	cipu0wdgsnq9u8st8m1lym0hq.com
	1ygx14u1vnf8hb1twhv8619h8ygr.net
Nymaim	ghhimbgbrpx.biz
	embonxn.info
Ramnit	ixajqhvaegcqgrhiwyv.com
	xykthadf.com

Several studies have focused specifically on collecting DNS datasets that alleviate the bias caused by various factors. For example, Kountouras *et al.* [36] implemented a system called Thales, that creates massive amounts of malicious domain names by distilling multiple freely available sources. Pearce *et al.* [37] developed a scalable, accurate, and ethical system, called Iris, that measures global name resolution and uses active manipulation to track the trends of domain names that evolve over time. Unfortunately, these systems cannot capture DGA-generated domains, which have extremely short lifetimes; hence, they require further improvements to handle DGAs.

III. PROPOSAL

In this article, we aim to identify malware-generated malicious domains from massive DNS queries. Table 1 shows

some examples of domain names generated by DGA malware. One challenge is that malicious domains in DGA malware survive only for extremely brief periods of time, which effectively restricts the set of features available for domain classification. The extensive observation of DNS traffic alleviates the restrictions on available features, but this effect clearly depends on the observed network scale. We address these problems by proposing an approach for identifying dynamically generated domains based on superficial analysis of those character strings. Note that natural language processing techniques cannot be directly applied to the superficial analysis of domain names because domain names are short, unstructured, and unsegmented character strings. This approach is based on the following observations: human-generated benign domain names tend to reflect the intent of their domain registrants, such as an organization, product, or content. In contrast, dynamically generated malicious domain names consist of meaningless character strings because conflicts with already registered domain names must be avoided; hence, there are discernible differences in the strings of dynamically generated and human-generated domain names. Actually, the major DGA malware such as Emotet, Mirai, and Ramnit that caused a pandemic crisis has this characteristic in domain names [6], [7].

Notably, the idea of using randomization to facilitate malware detection is not novel. Indeed, several studies based on this strategy have been reported [27], [28], [29], [30], [38], [39]. For example, Lin and Lee [38] proposed a method for decrypting malware communications for tracebacks to cybercriminals, while Wahab *et al.* [39] proposed a method for detecting compromised virtual machines by monitoring behavior at the hypervisor level. Other studies [27], [28], [29], [30] have attempted to apply ML or DL algorithms to distinguish between benign and malicious domains by noting the differences in character strings in the same manner as the approach proposed here. However, the cumbersome task of assigning labels to datasets and the discrepancies that arise between training and actual datasets have thus far prevented ML- and DL-based detection methods from exhibiting their full potential performance levels. A key point to emphasize is that our approach requires no prior information whatsoever regarding DGAs. Instead, it distinguishes between benign and malicious domains by determining whether or not a character string is meaningful. This approach mitigates dataset-related problems that degrade detection performance levels.

Figure 2 shows an overview of our proposed approach, which has four steps: (1) noise reduction, (2) subdomain selection, (3) dictionary-based estimation, and (4) Web-search-based estimation. In the following sections, we describe each of these steps in detail.

This proposed approach was initially introduced in our previous work [40]. In this article, we considerably extend the previous study by further enhancing the sophistication of each function, evaluating the approach from various perspectives through experiments, extensively discussing the experimental results, and comparing our approach with several published methods. Overall, this article presents a substantial evidence to demonstrate the effectiveness of our approach.

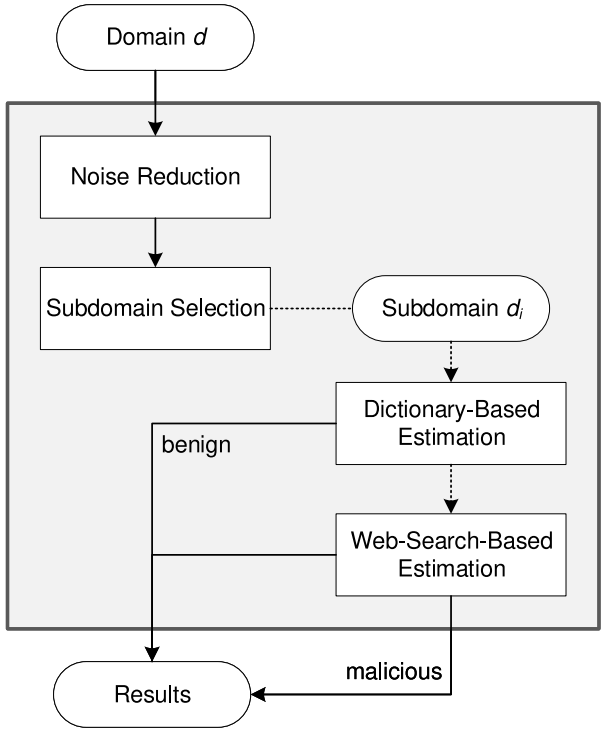

FIGURE 2. Overview of the proposed approach for identifying malicious domain names based on superficial analysis of their character strings.

A. NOISE REDUCTION

A query log for the input of our approach is a record of DNS queries for name resolution to RDNSs from machines on a network. Figure 3 shows an example of a query log for an RDNS with address 192.168.0.1, where lines labeled **Q** are requests for name resolution while lines labeled **R** are responses to these requests. As noted in Section II-A, a change in a callback destination by DGA malware is commonly accompanied by NXDOMAIN messages as the result of forward lookup. Thus, by restricting attention to specific DNS queries that meet both request of forward lookup and response of NXDOMAIN message, we can substantially reduce the number of domains requiring identification as either benign or malicious. Specifically, the underlined domain names, i.e., `www.kyutech.ac.jp` and `mail.comsoc.org`, are selected in this figure.

The first step is to identify benign domains via comparison with whitelists. The domains included in whitelists are those used by specific applications or services that cause massive NXDOMAIN messages, such as endpoint antivirus software and DNS-based lists (DNSBL). For example, as the name implies, DNSBL is a domain database that queries entries through the DNS protocol [41]. To investigate the domains `kyutech.ac.jp` and

```
Q1: 10-Jul-2019 15:00:31.265 client 192.168.10.240#62056 (1.0.20.10.in-addr.arpa): ↩
      query: 1.0.20.10.in-addr.arpa IN PTR + (192.168.0.1)
R1: 10-Jul-2019 15:00:44.346 client 192.168.10.240#62056: UDP: query: 1.0.20.10.in-addr.arpa IN PTR ↩
      response: NXDOMAIN +
Q2: 10-Jul-2019 15:00:53.470 client 192.168.20.120#44311 (www.kyutech.ac.jp): ↩
      query: www.kyutech.ac.jp IN A + (192.168.0.1)
R2: 10-Jul-2019 15:01:17.866 client 192.168.20.120#44311: UDP: query: www.kyutech.ac.jp IN A ↩
      response: NXDOMAIN +
Q3: 10-Jul-2019 15:01:53.467 client 192.168.20.120#55395 (www.ieee.org): ↩
      query: www.ieee.org IN A + (192.168.0.1)
R3: 10-Jul-2019 15:02:00.021 client 192.168.20.120#55395: UDP: query: www.ieee.org IN A ↩
      response: NOERROR + www.ieee.org. 300 IN CNAME e123.ieee.org.; e123.ieee.org. 300 IN A 10.0.0.1;
Q4: 10-Jul-2019 15:02:13.470 client 192.168.10.240#59283 (mail.comsoc.org): ↩
      query: mail.comsoc.org IN AAAA + (192.168.0.1)
R4: 10-Jul-2019 15:02:37.866 client 192.168.10.240#59283: UDP: query: mail.comsoc.org IN AAAA ↩
      response: NXDOMAIN +
Q5: 10-Jul-2019 15:03:04.053 client 192.168.30.100#51861 (sl._domainkey.kyutech.jp): ↩
      query: sl._domainkey.kyutech.jp IN TXT + (192.168.0.1)
R5: 10-Jul-2019 15:03:17.614 client 192.168.30.100#51861: UDP: query: sl._domainkey.kyutech.jp IN TXT ↩
      response: NOERROR + sl._domainkey.kyutech.jp. 86140 IN TXT "v=DKIM1;k=rsa;t=s;p=B3DQE ... ";
```

FIGURE 3. Example of a query log for an RDNS.

comsoc.org via Spamhaus, which is one of DNSBL services, a machine would attempt name resolution for the domain kyutech.ac.jp.sbl.spamhaus.org and comsoc.org.sbl.spamhaus.org. Since the domain names contain specific character strings in their high levels, they can be identified by a simple comparison. Next, domain names that violate DNS specifications are deemed to be the result of misconfiguration and are discarded [42]. The remaining domain names are considered possible DGA-generated name and are passed to subsequent functional units for further analysis.

B. SUBDOMAIN SELECTION

To efficiently identify domains, this step selects the subdomain d_i with the longest string from domain d, where i is an index of the subdomain level. For example, when domain d is xjjjvqpoh.com.ai, the largest string is xjjjvqpoh of sub-domain d_3. This step leverages the observations that DGAs generate relatively long domain names because conflicts with already registered domain names must be avoided and short domain names are more likely to be occupied by legitimate organizations.

C. DICTIONARY-BASED ESTIMATION

This step initially splits the subdomain d_i string into a word group w using dictionaries. The randomness of subdomain d_i is estimated from the features of word group w. If the result indicates that subdomain d_i is a meaningful string of characters, then domain d, which contains this subdomain, is determined to be a human-generated benign domain.

Six dictionaries are used for this step: one is an English dictionary and a corpus collected via Web crawling; the other five dictionaries are created by adding French, German, Russian, Spanish, and Japanese dictionaries. Considering the domain notation, we replace nonalphabetic representations with alphabetic representations in the dictionaries for these languages.

The subdomain d_i string is split into a word set w based on two conditions: (1) the number of words is the minimum, whereas the length of words is the maximum and (2) preference is given to words that are in the dictionary by differentiating with extreme selectivity using the following equation:

$$\mathcal{F}(d_i) = \arg\max_{w \in \mathbb{W}(d_i)} \frac{1}{n} \prod_{j=1}^{n} \mathcal{P}(w_j)$$
$$\mathcal{P}(w_j) = \begin{cases} 1 & (w_j \in \mathbb{D}) \\ 1/|\mathbb{D}|^{|w_j|} & (w_j \notin \mathbb{D}) \end{cases} \quad (1)$$

where $\mathbb{W}(d_i)$ is the set of all candidate segmentations of subdomain d_i, w is the candidate group of words $w_1, \ldots, w_j, \ldots, w_n$, $|w_j|$ is the length of word w_j, and $|\mathbb{D}|$ is the total number of words in dictionary $\mathbb{D}$. Furthermore, $\mathcal{P}(w_j)$ is the selectivity of word w_j, which is based on whether word w_j is in dictionary $\mathbb{D}$. The above first and second conditions correspond mainly to the $1/n$ and $\mathcal{P}(w_j)$ parts of this equation. This step selects the best result from the segmentations produced by the six dictionaries. If the result shows that $n = 1$, i.e., if sub-domain d_i is included in dictionary $\mathbb{D}$, then domain d is determined to be benign.

Since a human-generated domain string is characterized by a small number of long words, we estimate the randomness of subdomain d_i as follows:

$$y_\alpha = \sum_{k=1}^{n} u_k \mathcal{L}_k(w) \quad (2)$$

where u_k is a weight on function $\mathcal{L}_k(w)$, which rearranges words in descending order based on length and then gives the length of the k-th word in set w. For example, when word group w consists of kyutech, local, domain, and name, $\mathcal{L}_2(w)$ and $\mathcal{L}_4(w)$ output 6 and 4, which are the character lengths of domain and name. Note that the words that are not in the dictionary are assigned a length of 0. If the condition $y_\alpha > th_\alpha$ is satisfied, then domain d is determined to be benign.

TABLE 2. Differences between Web search results for benign and malicious subdomain strings.

Benign domains	Web search result
kyutech	175,000
comsoc	133,000
centrum24	66,400
incometaxindiaefiling	117,000
curapelanatureza	791,000
xn--sjqw6xkwbgyd9ay88l	789,000
Malicious domains	**Web search result**
7qvdqaw561dtasyi	0
pfnnwjoeuee	1
jlbroeji	1
cipu0wdgsnq9u8st8m1lym0hq	0
ghhimbgbrpx	0
ixajqhvaegcqgrhiwyv	1

D. WEB-SEARCH-BASED ESTIMATION

To compensate for dictionary deficiencies, this step estimates the randomness of a subdomain d_i by referring to the Web search result. Domain names that use languages without dictionaries, proper nouns, acronyms, initialisms, or Punycodes [43], for instance, cannot be identified as benign or malicious by the dictionary-based lexical analysis.

This step queries the Web search engine based on two conditions: (1) an exact match to the subdomain d_i string and (2) candidates for the th_β-th and subsequent Web pages to reduce the difference between the number of displayed hits and the number of actual Web pages. An example of such a query is shown below:

$$str = \text{“}d_i\text{”}\&num = th_\beta \tag{3}$$

Here, *str* indicates the character string d_i to be searched. The double quotes are an operator that returns Web pages that contain an exact match to the character string d_i, while *num* is an operator that returns the candidates for the th_β-th and subsequent Web pages. The ampersand symbol refers to the logical product of two operations. Note that character string d_i must be decoded beforehand when it is represented by Punycode, such as `xn-1lq68wkwbj6u` and `xn-sjqw6xkwbgyd9ay88l`.

Table 2 shows the differences between the Web search results for the benign and malicious subdomain strings. As is evident from the table, this metric reflects the benign and malicious nature of each domain. If the resulting number of Web pages associated with the string exceeds the threshold, $y_\beta > th_\beta$, then domain d is determined to be a human-generated benign domain; otherwise, domain d is considered a dynamically generated malicious domain.

This step dramatically mitigates the requirement for dictionaries in the previous step, dictionary-based estimation. This is because domain names in minor languages are identified by referring to the Web search result. For example, in a Japanese organization, our approach will ensure sufficient performance levels by only preparing Japanese and English dictionaries in the previous step. Meanwhile, this step is an extremely time-consuming process because of its reliance on the Web search results.

TABLE 3. Specifications of the machine used for implementation.

CPU	Xeon Silver 4110 (8core 2.10GHz)
GPU	NVIDIA GeForce RTX 2080 Ti (11GB GDDR6)
RAM	96GB DDR4-2666
SSD	Seq. Read and Write up to 560MB/sec and 530MB/sec
Kernel	Linux 3.10.0-957.1.3.el7.x86_64
Software	TensorFlow 2.0.0, CUDA Toolkit 10.0, cuDNN 7.6.5

Our strategy for addressing this issue is to substantially reduce the number of input domains for this last step by gradually identifying benign and malicious domains via the previous three steps, thereby yielding a major improvement in computational cost without a degradation in accuracy.

IV. EVALUATION

This section presents evidence to support the effectiveness of the proposed approach in detecting callbacks caused by DGA malware. We verify the identification accuracy of this approach from various perspectives and then extensively discuss its effectiveness and limitations in practical operation. Finally, we clarify the functional differences between our approach and other published methods via qualitative comparisons.

A. EXPERIMENTAL SETUP FOR LABELED DATASETS

For comparison with the proposed approach, we implemented three techniques for identifying malicious domains from only the domain name string based on [27], [28], and [29]. The specifications of the machine used for implementation are given in Table 3. The first technique uses bigram modeling with ML algorithms, whereas the second technique uses character-level domain modeling with DL algorithms. The third technique gives proper weight values to the character-level modeling by using attention mechanisms. Unlike our approach, these three implementations require training processes with datasets for identification.

Table 4 shows the datasets of benign and malicious domains used in our experiments. The malicious domains were generated by 19 types of DGAs, such as ChinAd, Conficker, Locky, NewGOZ, Nymaim, and Ramnit, whose algorithms were inferred through binary analysis [44]. The benign domains were the top 1,000,000 domains provided by Alexa [45]. For the training datasets of the three implementations, we randomly extracted 5% of the benign domains and 15% of the malicious domains. The remaining domains were used as the testing datasets. In the evaluation process, these benign and malicious datasets were used instead of DNS queries measured from operating networks because a strict comparison between our approach and the three other implementations would be difficult without correctly labeled domains. Furthermore, we disabled the first step, noise reduction, in our approach because these datasets did not have any information about DNS response.

In the proposed approach, we set the number of processes for parallelization to 10 because the processing for each input

TABLE 4. Numbers of benign and malicious domains in the datasets.

$D0$	$D1$	$D2$	$D3$	$D4$	$D5$	$D6$	$D7$	$D8$	$D9$
Alexa	ChinAd	Conficker	CoreBot	Fobber	Kraken	Locky	NewGOZ	Nymaim	PadCrypt
1000000	3000	3000	3000	3000	3000	3000	3000	3000	3000
$D10$	$D11$	$D12$	$D13$	$D14$	$D15$	$D16$	$D17$	$D18$	$D19$
Proslikefan	Pykspa	Qadars	Ramnit	Shiotob	Simda	Symmi	Tempedreve	Tinba	Vawtrak
3000	3000	3000	3000	3000	3000	3000	3000	3000	3000

TABLE 5. Experimental results.

	Recall	Precision
Truong et al. [27]	0.7915	0.5366
Anderson et al. [28]	0.8447	0.6462
Qiao et al. [29]	0.8585	0.6993
Our work	0.9960	0.9029

domain is completely independent. Additionally, we used words registered in the corpora [46], [47] and Aspell [48] as the dictionaries for the third step. Their total number of words was 8,000,000. The other parameters were set to the following values based on our experience: $u_1 = 2$, $u_2 = 1$, $u_3 = 0.25$, $th_\alpha = 15$, and $th_\beta = 50$. A large value was used for y_β based on the Web search results of `yahoo.com` and `bing.com`. The parameter optimization will be addressed in future work.

B. IDENTIFICATION ACCURACY FOR LABELED DATASETS

Two common metrics are employed to characterize the ability to identify benign and malicious domains: recall is the ratio of the number of correctly predicted malicious domains to the total number of actual malicious domains, and precision is the ratio of the number of correctly predicted malicious domains to the total number of predicted malicious domains.

The experimental results are presented in Table 5. These results indicate that our approach provides the highest level of performance, with recall and precision of 0.9960 and 0.9029, respectively. The other three implementations have lower recall and precision levels for the following reasons. First, the size of the training datasets was insufficient, and the features were biased. Second, identifying benign and malicious domains from character patterns alone has limitations. Our approach could conceivably eliminate these factors, which lower the accuracy levels, because it does not require prior knowledge of the training datasets and instead identifies domain names by superficially analyzing their character strings.

The numbers of misidentified domains are presented in Table 6. Our approach had a notably high identification ability for the 14 malware types, whereas Alexa, Conficker, Nymaim, Proslikefan, Pykspa, and Vawtrak had notable numbers of misidentifications. Specifically, the numbers of misidentified domains were 5189 and 102 for Alexa and Conficker, respectively.

We then checked the domain names generated by Conficker, which had random strings with lengths of 4 to 12 characters, and found that only domain names with 5 or fewer characters were misidentified. Specifically, the domains were misidentified because the approach could not distinguish between a benign domain containing an initialism or acronym and a dynamically generated malicious domain when the length of the domain name was short. In Nymaim, Proslikefan, Pykspa, and Vawtrak, the misidentification of domains resulted from randomly generated strings with approximately 6 characters that were meaningful words. Examples of such domains are `wouled.biz`, `olleman.com`, and `docket.com`.

Four types of misidentification occurred in Alexa: (a) domain names containing initialisms or acronyms, (b) domain names transcribed from nonalphabetic characters to alphabetic characters, (c) domain names with multiple numeric characters, and (d) domain names with random characters. The cause of the misidentification for the domains in (a) was the same as that for the Conficker dataset, whereas the domains in (b) and (c) were misidentified because such names were not included in the dictionaries. The domains in (d), such as `3lqjnuhra3xf585jgtkhk71e` `xuhu6yrkna.com` and `ctxxgxdnhctxxgxdnh.xyz`, were incorrectly identified as malicious by our approach, which focused solely on the character string of the domain name.

The evaluation results confirmed that the proposed approach achieves the highest accuracy despite the lack of advanced training: the recall and precision of our method reached 0.9960 and 0.9029, respectively.

C. EXPERIMENTAL SETUP FOR DNS QUERIES COLLECTED VIA CAMPUS NETWORKS

Figure 4 shows the layout of our campus network, which has two class B address blocks. This network consists of one core network, 135 access networks managed by 30 departments, and three wireless networks. We have managed only the core and wireless networks, including the connection points to the access networks. The total number of media access control (MAC) addresses observed at their connection points was approximately 9500. A total of 470 access points are placed in the three wireless networks that geographically cover most of our campus. The machines of more than 6000 employees and students, in addition to numerous visitors, routinely connect to the three wireless networks. The total number of machines connected to the three wireless networks was 6500 at a given point in time: 43% of them were iOS, 28% were Windows, 22% were Android, and 7% were macOS. As expected, there are no details of the 135 access networks outside our management areas. The dataset used for the experiments comprises DNS queries

TABLE 6. Numbers of misidentified domains in the datasets.

	$D0$	$D1$	$D2$	$D3$	$D4$	$D5$	$D6$	$D7$	$D8$	$D9$
Truong et al. [27]	33108	110	660	98	340	432	509	18	652	282
Anderson et al. [28]	22407	22	522	34	146	202	274	2	514	222
Qiao et al. [29]	17887	43	488	78	125	366	261	18	462	221
Our work	5189	0	102	0	0	2	1	0	25	0
	$D10$	$D11$	$D12$	$D13$	$D14$	$D15$	$D16$	$D17$	$D18$	$D19$
Truong et al. [27]	674	702	278	448	246	638	1586	632	314	1480
Anderson et al. [28]	576	580	286	218	150	626	1272	396	152	1326
Qiao et al. [29]	488	547	305	187	117	518	932	427	167	1101
Our work	19	21	0	0	0	3	1	0	0	17

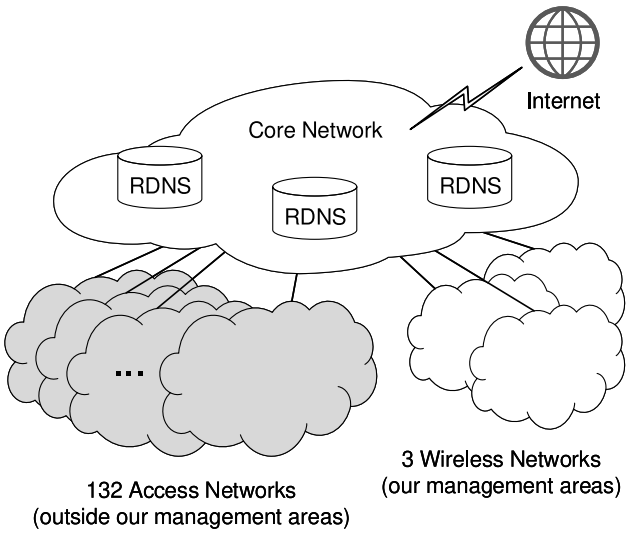

FIGURE 4. Layout of our campus network.

TABLE 7. Representative domain names included in whitelists.

(1) Our institution	
kyutech.ac.jp	kyutech.jp
(2) Spam detection	
spamhaus.org	uribl.com
spamcop.net	rfc-ignorant.org
support-intelligence.net	isipp.com
senderscore.com	dnswl.org
(3) Security products	
sophosxl.net	barracudacentral.org
trendmicro.com	mcafee.com
(4) Top 500 domains provided by Alexa	
google.com	linkedin.com
instagram.com	youtube.com
facebook.com	stackoverflow.com

observed on three RDNSs in the campus network during the one-month period beginning on December 1, 2019, which total approximately 65 GB in size. Among them, 3,304,505 queries satisfied both request of forward lookup and response of NXDOMAIN message.

For the parameters in our approach, we used values equivalent to those used for the evaluations discussed in Section IV-A. Meanwhile, we enabled the first step, noise reduction. This step has whitelists in which the total number of domains is 537, and some representative domains found in the whitelists are listed in Table 7. Because our campus network incorporates several operational mail servers, each of which has a spam filtering tool, domains accessed in the process of identifying spam are whitelisted. Other whitelisted domains include those of our institution itself and domains arising in inquiries made by security products or antivirus software. Notably, such domains result in a massive number of NXDOMAIN messages. In addition, we have registered the top 500 domains provided by Alexa with the whitelists to shorten the calculation time by excluding frequently accessed domains.

D. IDENTIFICATION ACCURACY FOR DNS QUERIES COLLECTED VIA CAMPUS NETWORKS

Table 8 shows the relationships among the number of input queries, number of unique domains, and calculation time for each step of our approach. The calculation time was 9.338 seconds for noise reduction, 9.338 seconds for subdomain selection, 20.665 seconds for dictionary-based estimation, and 12228.100 seconds for Web-search-based estimation. Thus, our approach was able to completely analyze a dataset with a total of 3,304,505 queries in 12267.521 seconds. Notably, the last step is an extremely time-consuming process because of its reliance on the results of Web searches. Our strategy for addressing this issue is to substantially reduce the number of input domains for the last step by focusing only on DNS queries that meet both request of forward lookup and response of NXDOMAIN message by gradually identifying benign and malicious domains via the previous three steps and by caching the search results for the same strings. The results show a major improvement in computational cost, thereby demonstrating the ability to handle massive DNS queries.

In this evaluation, 46954 queries were determined to be malicious by our approach, and these queries were mainly categorized into four types in terms of their causes. The first group comprised 6520 queries for 5006 unique domains that were dynamically generated by malware-infected machines. Examples of these queried domains are `umh7hqoxhrv3-0dmhzfk4.com`, `e6hmegdcjm-me4rtxlpqd-xh.com`, and `wmsbujwmnlxcin54wn9jxw5jewq.com`. The reasons for considering these domains as truly malicious are as follows: the domains consisting of meaningless character strings were queried by specific machines, and these queries were observed during a long time period without their causative communications.

The second group comprised 22714 queries with 6143 unique domains caused by Chrome or Chromium browsers.

TABLE 8. Relationship between the number of input queries, number of unique domains, and calculation time for each step in our approach.

	(1) Noise reduction	(2) Subdomain selection	(3) Dictionary-based estimation	(4) Web-search-based estimation	Results
Number of input queries	3304505	421883	421883	180599	46954
Number of unique domains	631926	27720	27720	16522	11388
Calculation time	9.418 s	9.338 s	20.665 s	12228.100 s	12267.521 s

TABLE 9. Qualitative comparison of our work with other well-known detection methods.

	(1) DGA malware detection	(2) Online detection	(3) Robust to encryption	(4) Network scale independent	(5) No need for a priori dataset
Soldo et al. [13]	Poor	Good	Good	Poor	Poor
Gu et al. [18]	Poor	Good	Poor	Good	Good
Bilge et al. [21]	Poor	Fair	Good	Good	Poor
Rahbarinia et al. [22]	Poor	Good	Good	Fair	Poor
Berger et al. [23]	Fair	Fair	Good	Poor	Fair
Wang et al. [24]	Good	Poor	Good	Poor	Good
Truong et al. [27]	Poor	Good	Good	Good	Poor
Anderson et al. [28]	Fair	Good	Good	Good	Poor
Qiao et al. [29]	Fair	Good	Good	Good	Poor
Li et al. [30]	Fair	Good	Good	Good	Poor
Our work	Good	Poor	Good	Good	Good

After startup, this type of Web browser sends a few DNS queries for three random-looking domains to check whether NXDOMAIN responses are hijacked in a particular network. In most cases, these queries are considered benign by the first step of noise reduction because their domains have the institutional suffix, e.g., `tyznhcicwa.kyutech.ac.jp`, or no suffix at all, e.g., `tyznhcicwa`. However, our performance levels are degraded when the domain search list is not overwritten via the dynamic host configuration protocol (DHCP) [49], specifically, when a machine connects to our campus network through virtual private network (VPN) tunneling and then a Chrome browser runs on the machine. In that scenario, if the domain search list of the machine set to `example.com` is not updated, the machine issues queries for domain names such as `tyznhcicwa.example.com`, `jxutrrw.example.com`, and `nlcngpby.example.com`. As a result, our approach falsely detects such queries as malicious because of the lack of a superficial difference.

The third group comprised 14221 queries for 218 unique domains associated with mail services. A mail server with an installed spam filtering tool, such as SpamAssassin, verifies the legitimacy of the sender through forward and reverse lookup. However, in some cases, NXDOMAIN responses are returned due to their name resolution being undefined at ADNSs. Examples of these queried domains are `net6-ip35.linkbg.com`, `188x235x148x209.static-business.saratov.ertelecom.ru`, and `static-ip-227-53-148-203.rev.dyxnet.com`. Note that the domains with an embedded IP address cannot be accurately identified from dictionary-based and Web-search-based analytical results because the embedded format must be interpreted.

The last group comprised 3499 queries with 21 unique domains due to various services. As a concrete example, some queries for domains such as tracking services, cloud services, and video distribution services, were erroneously detected. These causes are considered to be service down or misconfiguration.

The above three types of misidentification problems may be avoided by improving the noise reduction step. Specifically, the second and third groups were associated with unique query patterns and specific query sources, and some domain names in the last group had fixed character strings. We confirmed that over 95% of the errors were excluded by adding a filtering process based on those characteristics. Consequently, these results suggest the proposed approach achieves a high level of effectiveness for datasets collected via campus networks.

E. QUALITATIVE COMPARISON

Table 9 presents a qualitative comparison of the proposed approach and the ten previously published methods, which are representative examples of blacklist-based detection [13], DPI-based detection [18], reputation-based detection [21], behavior-based detection [22], [23], [24], ML-based detection [27], and DL-based detection [28], [29], [30]. In this comparison, we focus on the following five items: (1) DGA malware detection performance, (2) the ability to detect malware in real time, (3) robustness to encryption, (4) dependence on network scale, and (5) the need for training in advance using datasets or other prior knowledge. These items are chosen to compare detection performance and operational limitations, which are the practical issues of each method discussed in Section II-B, IV-B, and IV-D. Items (1) and (2) correspond to detection performance, items (3) and (4) to operational limitations, and item (5) to both of them. The evaluations conducted thus far indicate that our approach can detect the callbacks of DGA malware with high accuracy despite the lack of advance training and that the method is effective in actual operating networks. Meanwhile, ML- and DL-based methods [27], [28], [29], [30] are unable to exhibit high performance levels when the training datasets are insufficient in size and their features are biased. The performance

of our approach is not affected by the encryption commonly used to protect communications on the Internet because the method considers only DNS queries. Moreover, in contrast to the DNSMap of Berger *et al.* [23] and DBod of Wang *et al.* [24], our approach does not require observations from large-scale networks, such as ISPs. One drawback is the reliance on the results of Web searches, which makes the process of distinguishing between benign and malicious domains time consuming. Our strategy for addressing this issue is to substantially reduce the number of input domains for the fourth step, Web-search-based estimation, by gradually identifying benign and malicious domains via the previous three steps, thereby yielding a major improvement in computational cost without a degradation in accuracy.

The importance of encrypted communications in networks has spurred efforts to standardize protocols such as DNS over TLS (DoT) [50], which encapsulates DNS queries and their responses inside TLS communications to prevent eavesdropping and tampering. However, the only information required to distinguish between benign and malicious domains by our approach is the character strings of the domain names. Consequently, our approach is capable of operating solely referring to the system logs recorded by RDNSs and is thus entirely unaffected by DoT encryption.

In ML- and DL-based methods, the most important process for maintaining high performance levels is to ensure the training models are continuously updated while paying appropriate attention to concept drift [51]. Concept drift causes changes in the statistical properties predicted by a training model over time and leads to performance degradation. However, the task of aggregating strictly labeled datasets to update the model is unwieldy and cumbersome in practice. Our approach does not require prior dataset training, which means it is not susceptible to concept drift. Meanwhile, the approach includes a somewhat volatile metric because it relies on a Web search for each domain name in the fourth step, Web-search-based estimation. Despite this limitation, since DGA-generated domain names, which have extremely short lifetimes, are subject to search for only a short time period, the approach is able to detect malicious domains without being affected by search engine trends.

Over the past decade, a considerable number of article have been devoted to the study of adversarial machine learning [52], which exploits the vulnerability of ML- or DL-based methods to avoid detection. For example, Peck *et al.* [53] implemented CharBot, a simplistic character-based DGA that generates malicious domain names by randomly modifying two characters in well-known benign domain names, and Yun *et al.* [54] implemented Khaos, a novel DGA with high antidetection ability based on neural language models and Wasserstein generative adversarial networks. Unfortunately, the typical methods, which rely only on the learning results of domain names to make decisions, are inherently vulnerable to adversarial attacks by such DGAs; in contrast, our approach is highly resistant to such attacks because there is no learning process. Notably, our approach is not secure against all types of DGA malware. For example, in [55], Sood and Zeadally referred to Rovnix, which generates malicious domain names by concatenating specific words in a list. Such malicious domains have similar strings to benign domains, and, unfortunately, our approach cannot detect them. Examples of Rovnix-generated domain names in the literature, including `accelerateactress.in.net` and `accelerateaccountant.in.net`, are all identified as benign by the third step, dictionary-based estimation. Our approach significantly reduces the threat of major DGAs while ineffective against wordlist-based DGAs, leaving room for improvement in this respect. To solve this problem, as noted in the work of Pereira *et al.* [56] and Highnam *et al.* [57], it is necessary to consider the relations among words in the strings of domain names. However, these studies are now in the stage of embarking on ML- and DL-based detection with training datasets, and they still have not reached the stage of eliminating the dataset-related problems such as concept drift. As one of the first work to put forward DGA detection considering concept drift, we believe that this article holds important implications in guiding these studies to the next step in the future.

Despite the aforementioned problems, the proposed approach is more accurate than other methods and it also has high versatility because it is based solely on the character strings of domain names and does not require a learning process. Therefore, we expect that this approach will help to dramatically improve the performance of other published studies if it is incorporated into their techniques.

V. CONCLUSION

In this article, to identify dynamically generated malicious domains, we proposed an approach for estimating the randomness of domain names by superficially analyzing their character strings. This approach is based on the following observations: human-generated benign domain names tend to reflect the intent of their domain registrants, such as an organization, product, or content. In contrast, dynamically generated malicious domain names consist of meaningless character strings because conflicts with already registered domain names must be avoided; hence, there are discernible differences in the strings of dynamically generated and human-generated domain names. The concept of using randomization to facilitate malware detection has been advocated in previous studies, whereas our approach distinguishes between benign and malicious domains using only the character string of the domain name and does not require any prior knowledge about DGAs.

In the experiments conducted as part of this study, the results showed that our approach could achieve recall and precision as high as 0.9960 and 0.9029, respectively, when used with labeled datasets. We also achieved a high level of effectiveness for datasets collected via a campus network. Taken together, these results suggest that malware-infected machines can be swiftly identified and removed from networks using DNS queries for detected malicious

domains as triggers. Our approach contributes to dramatically improving network security by providing the ability to address various malware encroachments.

The novel characteristics of our strategy have ensured three levels of effectiveness. First, because our approach is capable of operating solely by referring to the system logs recorded by RDNSs, its applicability is not restricted to networks with unencrypted communications. Second, in contrast to ML- and DL-based detection techniques, our approach eliminates the problem of datasets characterized by concept drift, which facilitates the retention of high performance levels. Finally, our approach may be used in a wide variety of scenarios because of its high versatility. Indeed, we expect it to be possible to incorporate certain aspects of our approach into other published methods to substantially improve their malware detection performance.

In our future work, we intend to evaluate the efficacy of the proposed approach for communications observed over long time periods in large-scale networks. We also hope to investigate the resulting impact on their malware detection performance by combining our technique with other methods.

REFERENCES

[1] J. A. Lewis. *Economic Impact of Cybercrime—No Slowing Down, 2018*. Accessed: Aug. 1, 2020. [Online]. Available: https://www.csis.org/analysis/economic-impact-cybercrime

[2] Y. Fu *et al.*, "Stealthy domain generation algorithms," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1430–1443, Jul. 2017.

[3] Z. Chen, M. Roussopoulos, Z. Liang, Y. Zhang, Z. Chen, and A. Delis, "Malware characteristics and threats on the Internet ecosystem," *J. Syst. Softw.*, vol. 85, no. 7, pp. 1650–1672, 2012.

[4] T. Nelms, R. Perdisci, and M. Ahamad, "ExecScent: Mining for new C&C domains in live networks with adaptive control protocol templates," in *Proc. USENIX Conf. Security Symp.*, 2013, pp. 589–604.

[5] Cisco Systems, Inc. *Cisco Annual Cybersecurity Report 2018*. Accessed: Aug. 1, 2020. [Online]. Available: https://www.cisco.com/c/en/us/products/security/cybersecurity-reports.html

[6] Check Point Software Technologies Ltd. *February 2020's Most Wanted Malware: Increase in Exploits Spreading the Mirai Botnet to IoT Devices*. Accessed: Aug. 1, 2020. [Online]. Available: https://www.checkpoint.com/press/2020/february-2020s-most-wanted-malware-increase-in-exploits-spreading-the-mirai-botnet-to-iot-devices/

[7] Network Security Research Lab. *DGA—Netlab OpenData Project*. Accessed: Aug. 1, 2020. [Online]. Available: https://data.netlab.360.com/dga/

[8] R. Vinayakumar, K. P. Soman, P. Poornachandran, and S. S. Kumar, "Evaluating deep learning approaches to characterize and classify the DGAs at scale," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1265–1276, 2018.

[9] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Min. Knowl. Discovery*, vol. 30, no. 4, pp. 964–994, 2016.

[10] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data—AI integration perspective," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 8, 2019, doi: 10.1109/TKDE.2019.2946162.

[11] D. Kim, "Potential risk analysis method for malware distribution networks," *IEEE Access*, vol. 7, pp. 185157–185167, 2019.

[12] Y. Cai, G. O. M. Yee, Y. X. Gu, and C.-H. Lung, "Threats to online advertising and countermeasures: A technical survey," *ACM Digit. Threats Res. Pract.*, vol. 1, no. 2, p. 11, 2020.

[13] F. Soldo, A. Le, and A. Markopoulou, "Blacklisting recommendation system: Using spatio-temporal patterns to predict future attacks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 7, pp. 1423–1437, Aug. 2011.

[14] J. Freudiger, E. De Cristofaro, and A. E. Brito, "Controlled data sharing for collaborative predictive blacklisting," in *Proc. Int. Conf. Detect. Intrusions Malware Vulnerability Assess.*, 2015, pp. 327–349.

[15] S. Špaček, M. Laštovicka, M. Horák, and T. Plesník, "Current issues of malicious domains blocking," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Service Manag.*, 2019, pp. 551–556.

[16] S. Špaček, M. Laštovicka, M. Horák, and T. Plesník, "DNS firewall data visualization," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Service Manag.*, 2019, pp. 743–744.

[17] P. Vixie *et al.*, "DNS response policy zones (RPZ)," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-vixie-dnsop-dns-rpz-00, 2018.

[18] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: Detecting malware infection through IDS-driven dialog correlation," in *Proc. USENIX Conf. Security Symp.*, 2007, pp. 167–182.

[19] T. J. Parvat *et al.*, "Performance improvement of deep packet inspection for intrusion detection," in *Proc. IEEE Glob. Conf. Wireless Comput. Netw.*, 2014, pp. 224–228.

[20] J. Su, S. Chen, B. Han, C. Xu, and X. Wang, "A 60Gbps DPI prototype based on memory-centric FPGA," in *Proc. ACM SIGCOMM Conf.*, 2016, pp. 627–628.

[21] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A passive DNS analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Security*, vol. 16, no. 4, pp. 1–14, 2014.

[22] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Efficient and accurate behavior-based tracking of malware-control domains in large ISP networks," *ACM Trans. Privacy Security*, vol. 19, no. 2, pp. 1–4, 2016.

[23] A. Berger, A. D'Alconzo, W. N. Gansterer, and A. Pescapé, "Mining agile DNS traffic using graph analysis for cybercrime detection," *Comput. Netw.*, vol. 100, pp. 28–44, May 2016.

[24] T.-S. Wang, H.-T. Lin, W.-T. Cheng, and C.-Y. Chen, "DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis," *Comput. Security*, vol. 64, pp. 1–15, Jan. 2017.

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

[26] M. Zago, M. G. Pérez, and G. M. Pérez, "UMUDGA: A dataset for profiling DGA-based botnet," *Comput. Security*, vol. 92, May 2020, Art. no. 101719.

[27] D.-T. Truong and G. Cheng, "Detecting domain-flux botnet based on DNS traffic features in managed network," *Security Commun. Netw.*, vol. 9, no. 14, pp. 2338–2347, 2016.

[28] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: Adversarially-tuned domain generation and detection," in *Proc. ACM Workshop Artif. Intell. Security*, 2016, pp. 13–21.

[29] Y. Qiao, B. Zhang, W. Zhang, A. K. Sangaiah, and H. Wu, "DGA domain name classification method based on long short-term memory with attention mechanism," *Appl. Sci.*, vol. 9, no. 20, pp. 1–14, 2019.

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

[31] R. Sivaguru, C. Choudhary, B. Yu, V. Tymchenko, A. C. A. Nascimento, and M. De Cock, "An evaluation of DGA classifiers," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 5058–5067.

[32] B. Yu, D. L. Gray, J. Pan, M. De Cock, and A. C. A. Nascimento, "Inline DGA detection with deep networks," in *Proc. IEEE Int. Conf. Data Min. Workshops*, 2017, pp. 683–692.

[33] A. S. Iwashita and J. P. Papa, "An overview on concept drift learning," *IEEE Access*, vol. 7, pp. 1532–1547, 2019.

[34] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019.

[35] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, "TESSERACT: Eliminating experimental bias in malware classification across space and time," in *Proc. USENIX Security Symp.*, 2019, pp. 729–746.

[36] A. Kountouras *et al.*, "Enabling network security through active DNS datasets," in *Proc. Int. Symp. Res. Attacks Intrusions Defenses*, 2016, pp. 188–208.

[37] P. Pearce *et al.*, "Global measurement of DNS manipulation," in *Proc. USENIX Security Symp.*, 2017, pp. 307–323.

[38] W. Lin and D. Lee, "Traceback attacks in cloud—Pebbletrace botnet," in *Proc. Int. Conf. Distrib. Comput. Syst. Workshops*, 2012, pp. 417–426.

[39] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Optimal load distribution for the detection of VM-based DDoS attacks in the cloud," *IEEE Trans. Services Comput.*, vol. 13, no. 1, pp. 114–129, Jan./Feb. 2020.
[40] A. Satoh, Y. Nakamura, D. Nobayashi, and T. Ikenaga, "Estimating the randomness of domain names for DGA bot callbacks," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1378–1381, Jul. 2018.
[41] J. Levine, "DNS blacklists and whitelists," Internet Eng. Task Force, Fremont, CA, USA, Rep. RFC 5782, 2010.
[42] P. Mockapetris, "Domain names—Implementation and specification," Internet Eng. Task Force, Fremont, CA, USA, Rep. RFC 1035, 1987.
[43] A. Costello, "Punycode: A bootstring encoding of unicode for internationalized domain names in applications (IDNA)," Internet Eng. Task Force, Fremont, CA, USA, Rep. RFC 3492, 2003.
[44] J. Bader. *Some Results of My DGA Reversing Efforts*. Accessed: Aug. 1, 2020. [Online]. Available: https://github.com/baderj/domain_generation_algorithms
[45] Hacker Target Pty Ltd. *Download Top 1 Million Sites*. Accessed: Aug. 1, 2020. [Online]. Available: https://hackertarget.com/top-million-site-list-download/
[46] P. Norvig. *Natural Language Corpus Data: Beautiful Data*. Accessed: Aug. 1, 2020. [Online]. Available: http://norvig.com/ngrams/
[47] Linguatools. *Wikipedia Monolingual Corpora*. Accessed: Aug. 1, 2020. [Online]. Available: https://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/
[48] K. Atkinson. *GNU Aspell*. Accessed: Aug. 1, 2020. [Online]. Available: http://aspell.net
[49] B. Aboba and S. Cheshire, "Dynamic host configuration protocol (DHCP) domain search option," Internet Eng. Task Force, Fremont, CA, USA, Rep. RFC 3397, 2002.
[50] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for DNS over transport layer security (TLS)," Internet Eng. Task Force, Fremont, CA, USA, Rep. RFC 7858, 2016.
[51] M. D. Lange *et al.*, "A continual learning survey: Defying forgetting in classification tasks," 2020. [Online]. Available: arXiv:1909.08383.
[52] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018.
[53] J. Peck *et al.*, "CharBot: A simple and effective method for evading DGA classifiers," *IEEE Access*, vol. 7, pp. 91759–91771, 2019.
[54] X. Yun, J. Huang, Y. Wang, T. Zang, Y. Zhou, and Y. Zhang, "Khaos: An adversarial neural network DGA with high anti-detection ability," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2225–2240, 2020.
[55] A. K. Sood and S. Zeadally, "A taxonomy of domain-generation algorithms," *IEEE Security Privacy*, vol. 14, no. 4, pp. 46–53, Jul./Aug. 2016.
[56] M. Pereira, S. Coleman, B. Yu, M. De Cock, and A. Nascimento "Dictionary extraction and detection of algorithmically generated domain names in passive DNS traffic," in *Proc. Int. Symp. Res. Attacks Intrusions. Defenses*, 2018, pp. 295–314.
[57] K. Highnam, D. Puzio, S. Luo, and N. R. Jennings, "Real-time detection of dictionary DGA network traffic using deep learning," 2020. [Online]. Available: arXiv:2003.12805.

AKIHIRO SATOH received the Ph.D. degree in information sciences from Tohoku University. He is currently an Assistant Professor with the Information Science and Technology Center, Kyushu Institute of Technology, Japan. His research interests include network operation and network security. He is a member of the IEICE and IPSJ.

YUTAKA FUKUDA (Member, IEEE) received the D.E. degree in computer sciences from the Kyushu Institute of Technology, Japan, where he is currently an Associate Professor with the Information Science and Technology Center. His research interests include performance evaluation of computer networks, wireless networks, and transport protocols. He is a member of the IEICE and IPSJ.

TOYOHIRO HAYASHI received the D.E. degree in computer sciences from the Kyushu Institute of Technology, Japan, where he is currently an Assistant Professor with the Information Science and Technology Center. His research interests include development of information systems for education, computer vision, and robotics. He is a member of the IEICE and IPSJ.

GEN KITAGATA received the D.E. degree in information sciences from Tohoku University, Japan, where he is currently an Associate Professor with the Research Institute of Electrical Communication. His research interests include agent-based computing, intelligent networking, and communication systems. He is a member of the IEICE and IPSJ.