IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 19, 2019, accepted September 23, 2019, date of publication September 27, 2019,
date of current version October 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944203

A Cause-Based Classification Approach for Malicious DNS Queries Detected Through Blacklists

AKIHIRO SATOH[1], YUTAKA NAKAMURA[1], YUTAKA FUKUDA[1], (Member, IEEE), KAZUTO SASAI[2], AND GEN KITAGATA[3]

[1]Kyushu Institute of Technology, Kitakyushu 804-8550, Japan
[2]Graduate School of Science and Engineering, Ibaraki University, Hitachi 316-8511, Japan
[3]Research Institute of Electrical Communication, Tohoku University, Sendai 980-8577, Japan

Corresponding author: Akihiro Satoh (satoh@isc.kyutech.ac.jp)

This work was supported by the MIC/SCOPE under Grant 192210001.

ABSTRACT Some of the most serious security threats facing computer networks involve malware. To prevent this threat, administrators need to swiftly remove the infected machines from their networks. One common way to detect infected machines in a network is by monitoring communications based on blacklists. However, detection using this method has the following two problems: no blacklist is completely reliable, and blacklists do not provide sufficient evidence to allow administrators to determine the validity and accuracy of the detection results. Therefore, simply matching communications with blacklist entries is insufficient, and administrators should pursue their detection causes by investigating the communications themselves. In this paper, we propose an approach for classifying malicious DNS queries detected through blacklists by their causes. This approach is motivated by the following observation: a malware communication is divided into several transactions, each of which generates queries related to the malware; thus, surrounding queries that occur before and after a malicious query detected through blacklists help in estimating the cause of the malicious query. Our cause-based classification drastically reduces the number of malicious queries to be investigated because the investigation scope is limited to only representative queries in the classification results. In experiments, we have confirmed that our approach could group 388 malicious queries into 3 clusters, each consisting of queries with a common cause. These results indicate that administrators can briefly pursue all the causes by investigating only representative queries of each cluster, and thereby swiftly address the problem of infected machines in the network.

INDEX TERMS Malware, blacklist, domain name system, network security, machine learning.

I. INTRODUCTION

Some of the most serious security threats facing computer networks involve malware. Cyber-criminals use malware-infected machines to undertake malicious activities such as stealing confidential information, spreading malware to additional machines, and phishing to an organization. According to a recent McAfee report [1], over 300,000 new forms of malware are created each day, and the global annual cost may be as much as $600 billion; thus, it is imperative to establish security mechanisms to protect against malware.

In order to prevent damage from malware, administrators need to swiftly identify and remove the infected machines that reside in their networks. One common way to detect infected machines in a network is by monitoring communications based on blacklists. This method detects the suspected machines of infecting malware by matching communications with blacklist entries. To improve the detection capability of this method, several studies have attempted to automatically update blacklist entries by using machine learning techniques [2], [3]. However, detection using this method has the following two problems [4], [5]: (1) no blacklist is completely reliable, and (2) blacklists do not provide sufficient evidence to allow administrators to determine the validity and accuracy of the detection results. Therefore, simply matching

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

communications with blacklist entries is insufficient, and administrators should pursue their detection causes by investigating the communications themselves [6].

In this paper, we propose a novel approach for classifying malicious DNS (domain name system) queries detected through blacklists by their causes. We focus on the DNS because domain name resolution always occurs prior to malware communications and the name resolution is an unencrypted interaction. This approach is motivated by the following important observation: a malware communication is divided into several transactions, each of which generates queries related to the malware; thus, surrounding queries that occur before and after a malicious query detected through blacklists help in estimating the cause of the malicious query. Most previous work [7]–[10] adopted the superficial similarity of character strings and hierarchical structures in domain names. Unlike such superficial classification, our cause-based classification drastically reduces the number of malicious DNS queries to be investigated because the investigation scope is limited to only representative queries in the classification results. Through experiments using DNS queries observed on a campus network, we have confirmed that our approach could group 388 malicious queries into 3 clusters, each consisting of queries with a common cause. These results indicate that administrators can briefly pursue all the causes by investigating only representative queries of each cluster, and thereby swiftly address the problem of infected machines in the network.

The remainder of this paper is organized as follows: In Section II, we review the related studies and their limitations. In Section III, we propose a cause-based classification approach for malicious DNS queries detected through blacklists. We describe experiments conducted to analyze the effectiveness of our approach in classifying malicious DNS queries in Section IV. Finally, we summarize our conclusions and future work in Section V.

II. RELATED WORK

Soldo *et al.* [11] proposed a method for significantly improving blacklists based on previous attack logs provided by multiple contributors. Meanwhile, AutoBLG developed by Sun *et al.* [2] and Segugio developed by Rahbarinia *et al.* [3] are systems that automatically generate new blacklists from existing ones. The difference between these is that AutoBLG uses the results of web crawling, whereas Segugio uses the results of monitoring passive traffic for automatic generation. Špaček *et al.* [12], [13] developed a DNS firewall system that blocks communications from the protected network to malicious domains on the outside network. This system uses the DNS RPZ (response policy zones) technology [14] for advanced domain blacklisting. Studies that produce more sophisticated blacklists are frequently conducted and still constitute the core of network threat defense strategy.

Kheir *et al.* [4] showed that blacklists typically contain a considerable number of errors. These errors are due to domains containing mixed benign and malicious codes, such as in the cases of cloud computing services, advertising network services, and dynamic DNS services. Automatically generated blacklists simply exacerbate this problem. They attempted to improve the detection accuracy by cross-checking domains on multiple blacklists. However, each blacklist has a specific coverage, so cross-checking them greatly narrows the effective coverage.

Kührer *et al.* [5] evaluated the effectiveness of 19 types of blacklists by considering such factors as unregistered domains, parking domains, and sinkhole domains. Their evaluation used only datasets prelabeled as benign or malicious. This is due to the fact that the investigation scope expands in proportion to the number of detections, which complicates determining the validity and accuracy of detection results, as was also pointed out by [15].

Gomez *et al.* [6] described that the errors due to the fallacious severity in detection systems interfere with normal activities; accordingly, administrators should pursue the detection causes by investigating the communications themselves. To address this problem, they developed a threat analysis console named THACO that visualizes DNS queries with multiple heterogeneous network threat intelligence sources. This system focuses on visualization while our approach focuses on classification, both of which assist administrators in efficiently analyzing malicious DNS queries and swiftly pursuing their causes.

Since DNS queries accurately reflect various activities [16], many anomaly detection methods for DNS queries have been proposed over the past decade. For example, Cui *et al.* [17] explored various data mining algorithms for obtaining useful patterns from an enormous volume of fast evolving DNS queries. Robberechts *et al.* [18] designed and implemented an anomaly detection system named QLAD that is applicable to the high volume and specific nature of queries to the top-level DNS servers. In experiments, QLAD found several anomalies of the sort that are of interest to registry operators, such as domain enumerations and DoS attacks. Li *et al.* [19] established a machine learning framework to handle DGA (domain generation algorithm) malware threats. A DGA is a technique to hide the callback communications from infected machines to their C&C (command-and-control) server. This framework leverages the behaviors of DNS queries for detecting infected machines because C&C domains have different characteristics when compared with other domains. Besides the above work, various methods based on query behaviors have been proposed for detecting the certain types of threats, such as botnets [20], advanced persistent threat attacks [21], and water torture attacks [22].

Some previous studies have focused on analyzing DNS queries and responses like our approach. Wang *et al.* [23] developed a system called DBod that detects and classifies infected machines on the basis of statistical similarity between query behaviors. However, DBod is specific to DGA malware and cannot be adapted to handle other kinds of malware. Berger *et al.* [7] developed a system called DNSMap that discovers potentially compromised machines on the basis

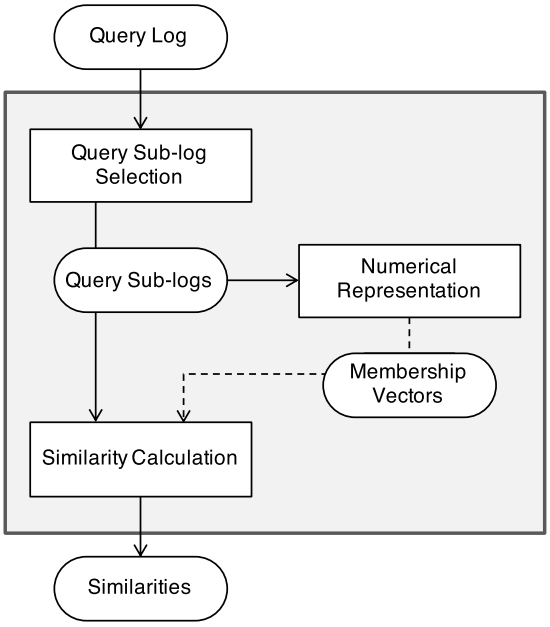


FIGURE 1. Overview of the cause-based classification approach for malicious DNS queries.

of rapidly changing DNS queries. DNSMap derives the similarity of character strings in domain names by considering their hierarchical structure, but the superficial similarity is insufficient for well classifying malicious queries by their causes. Not only DNSMap but also most other work [8]–[10] has adopted the superficial similarity of domain names.

III. PROPOSAL

In this paper, we propose a cause-based classification approach for malicious DNS queries detected through blacklists. This approach is motivated by the fact that a malware communication is divided into several transactions, each of which generates queries related to the malware, so the surrounding queries that occur before and after a malicious query detected through blacklists help in estimating the cause of the malicious query. By numerically comparing their surrounding queries, the approach is able to classify malicious queries by their causes. Unlike conventional classification, which is based on the superficial similarity of character strings in domain names, our cause-based classification can efficiently analyze malware communications, allowing measures for the infected machines in the network to be taken swiftly. General classification techniques in natural language processing could probably be applied by regarding a malicious query and surrounding queries as words. Le *et al.* [24] proposed Doc2Vec, which classifies various documents by using co-occurrences between words. Unfortunately, the performance deteriorates due to the influence of queries irrelevant to the classification. In contrast, our approach weights queries based on insight about malware communications.

Figure 1 shows an overview of the proposed approach, which has three functions: (1) Query Sub-log Selection, (2) Numerical Representation, and (3) Similarity Calculation. The following sections describe a query log and each of these functions in detail.

The proposed approach was initially introduced in our previous work [25]. We significantly extend the previous work by sophisticating each function further, evaluating the approach from various perspectives through experiments, and deeply discussing about the experimental results. Specially, this paper presents a lot of evidence to prove the effectiveness of our approach.

A. QUERY LOG

A query log for the input of our approach is a record of queries for domain name resolution to an RDNS (recursive DNS) server from machines on a network. Figure 2 shows an example of a query log for an RDNS server with address 192.168.0.1. In the query log, each query has values such as a timestamp, source address, and queried domain name along with class and type. In particular, note that domain names in the query log are shortened to and replaced with primary domain names. A primary domain is the highest-level domain name given to a registrar [26]. For example, the primary domain names for `www.ieee.org` and `smtp.kyutech.ac.jp` would respectively be `ieee.org` and `kyutech.ac.jp`.

B. QUERY SUB-LOG SELECTION

This function detects malicious queries from a query log through comparison with blacklists. Then, queries before and after a malicious query are selected from the query log, forming a query sub-log. Note that a query sub-log contains some queries related to malware communications.

A query x_n in the query log is considered to be malicious if the domain name in query x_n matches the entry in blacklist L_B. The function then selects all the queries with the same source address as malicious query x_n that occur within t_α seconds of either before or after malicious query x_n. These queries constitute the query sub-log X_n. This step is conducted because these queries in query sub-log X_n help in estimating the cause of malicious query x_n. Finally, the output of the function is set $\mathbb{X}$, which comprises the N number of query sub-logs, where N denotes the number of malicious queries detected in the query log.

C. NUMERICAL REPRESENTATION

This function attempts to numerically represent queries based on their co-occurrences by using two machine learning techniques: Word2Vec [27] and soft clustering with Gaussian mixture models [28]. This step extracts effective features from the enormous number of queries included in all the query sub-logs.

First, the function applies a Word2Vec model to all the query sub-logs $\mathbb{X}$ to create a distributed representation based on co-occurrences between queries. A distributed representation is to associate one data record with one point in multi-dimensional space, and Word2Vec, which has drawn considerable attention in the field of natural

```
11-Jan-2018 05:00:53.265 queries: info: client 192.168.10.240#35704 ↩
        (10.21.241.10.in-addr.arpa): query: 10.21.241.10.in-addr.arpa IN PTR + (192.168.0.1)
11-Jan-2018 05:00:53.467 queries: info: client 192.168.20.120#54171 ↩
        (smtp.kyutech.ac.jp): query: smtp.kyutech.ac.jp IN SOA + (192.168.0.1)
11-Jan-2018 05:00:53.470 queries: info: client 192.168.20.120#54311 ↩
        (ieeeaccess.ieee.org): query: ieeeaccess.ieee.org IN A + (192.168.0.1)
11-Jan-2018 05:00:53.470 queries: info: client 192.168.10.240#49193 ↩
        (ieeexplore.ieee.org): query: ieeexplore.ieee.org IN AAAA + (192.168.0.1)
11-Jan-2018 05:00:54.053 queries: info: client 192.168.30.100#54015 ↩
        (analytics.google.com): query: analytics.google.com IN TXT + (192.168.0.1)
11-Jan-2018 05:00:54.102 queries: info: client 192.168.10.120#33010 ↩
        (ns.isc.kyutech.ac.jp): query: ns.isc.kyutech.ac.jp IN NS + (192.168.0.1)
```

FIGURE 2. Example of a query log for an RDNS server.

language processing, expresses the features of each word as a vector based on the assumption that each word in a sentence has a strong relation with its surrounding words. We modify Word2Vec to change its focus from words in a sentence to queries in a query sub-log as follows: (1) we replace words with queried domain names, and (2) whereas the conventional Word2Vec algorithm uses the distance between words in a sentence to measure co-occurrences, we use instead the time interval between queries in a query sub-log, which is restricted to within t_β seconds.

Next, the function applies soft clustering with Gaussian mixture models to the distributed representation of queries. Soft clustering yields the probability that each data point belongs to each cluster. Because each cluster comprises queries with similar co-occurrences in the clustering results, a cluster can be expected to indicate a transaction in a malware communication. Finally, the output of the function is the membership vector of each query, written as follows:

$$\vec{p}(x_i) = \left(p(c_1|x_i), \cdots, p(c_m|x_i), \cdots, p(c_M|x_i) \right) ,$$

where M is the number of clusters and $p(c_m|x_i)$ is the probability that query x_i belongs to cluster c_m.

D. SIMILARITY CALCULATION

This function calculates the feature vector from the membership vectors of the queries found in each query sub-log. By comparing the similarity of feature vectors according to their cosine distance, the function achieves cause-based classification latently indicated by the malicious queries and their surrounding queries. Note that we emphasize queries commonly appearing in multiple query sub-logs based on the following insight about malware communications: infected machines in a same malware family repeatedly communicate with a same malicious domain group.

The membership vectors of the queries found in each query sub-log imply the transactions constituting a malware communication. In query sub-logs, the similarities of transactions are strongly dependent on the similarities of causes. Thus, the feature vector for each query sub-log is derived from the weighted sum of the membership vectors of the queries found in each query sub-log, as follows:

$$\vec{X}_n = \sum\nolimits_{x_i \in X_n} w_\alpha(x_i)\ w_\beta(x_i)\ w_\gamma(x_i)\ \vec{p}(x_i)\ .$$

Here, $w_\gamma = 0$ if the domain name for query x_i is included in whitelist L_W; otherwise, $w_\gamma = 1$. From the above insight concerning malware communications, the weights w_α and w_β for query x_i are respectively defined as

$$w_\alpha(x_i) = \frac{|\mathcal{F}_{addr}(x_i, \mathbb{X}) \cap \mathcal{F}_{name}(x_i, \mathbb{X})|}{|\mathcal{F}_{addr}(x_i, \mathbb{X})|}$$

and

$$w_\beta(x_i) = \frac{|\mathcal{F}_{list}(x_i, \mathbb{X}) \cap \mathcal{F}_{name}(x_i, \mathbb{X})|}{|\mathcal{F}_{list}(x_i, \mathbb{X})|}$$

where $\mathcal{F}_{name}(x_i, \mathbb{X})$, $\mathcal{F}_{addr}(x_i, \mathbb{X})$, and $\mathcal{F}_{list}(x_i, \mathbb{X})$ are differing subsets of set $\mathbb{X}$. $\mathcal{F}_{name}(x_i, \mathbb{X})$ is a set formed from query sub-logs including a query with the same domain name as query x_i; $\mathcal{F}_{addr}(x_i, \mathbb{X})$ is a set formed from query sub-logs that consist of queries with the same source address as query x_i; $\mathcal{F}_{list}(x_i, \mathbb{X})$ is a set formed from query sub-logs that are detected by the same blacklist entry as query sub-log X_n including query x_i. Additionally, $|\cdot|$ indicates the number of set elements. In short, w_α and w_β give high weights to the following two types: (1) queries to the same domain that repeatedly occur from a machine, and (2) queries to the same domain that repeatedly occur from machines detected by a blacklist entry. Finally, the output of the function is the similarities of malicious queries calculated by comparing the feature vectors of query sub-logs on the basis of their cosine distance.

IV. EVALUATION

In this section, we present our evaluation of the effectiveness of the proposed approach based on experiments using DNS queries observed on a campus network. This evaluation focuses primarily on the classification accuracy of malicious queries and the efficiency of the analysis of their causes. We first describe the experimental setup in Section IV-A and then discuss the experimental results in the following sections.

A. EXPERIMENTAL SETUP

Figure 3 shows the layout of our campus network. The campus network with two class B address blocks consists of 132 access networks managed by 31 departments and two wireless networks. We have managed only the core and wireless networks, including the connection points to

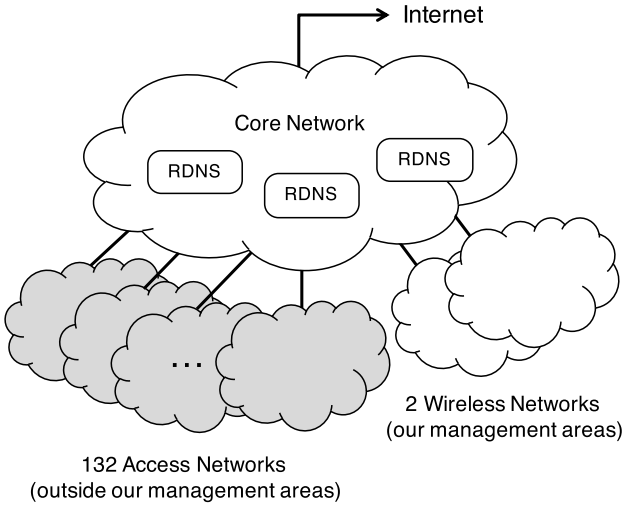


FIGURE 3. Layout of our campus network.

TABLE 1. Sources for blacklist L_B and whitelist L_W.

Blacklist L_B	Acquisition date	Number of entries
DNS-BH [29]	Mar 1, 2018	18,140
hpHosts [30]	Mar 1, 2018	745,338
abuse.ch [31]	Mar 1, 2018	2,308

Whitelist L_W	Acquisition date	Number of entries
Alexa [32]	Mar 1, 2018	1,000,000

TABLE 2. Specifications of a machine for implementation.

CPU	Intel(R) Core(TM) i5-4460 3.20GHz
GPU	NVIDIA GeForce GTX 750 Ti
RAM	DDR3-1600 16GB
SSD	Seq. Read and Write up to 770MB/sec and 580MB/sec
Kernel	Linux 3.10.0-693.11.1.el7.x86_64
Software	TensorFlow 1.3.0, CUDA Toolkit 8.0 with cuDNN 6.0

the access networks. The total number of MAC addresses observed at their connection points was about 9000. A total of 337 access points are placed in the two wireless networks that geographically cover most of our campus. The machines of more than 6000 employees and students in addition to many visitors connect to the two wireless networks. The total number of machines connected to the two wireless networks was 6500 at a given point in time; 56% of them were iOS, 20% were Windows, 20% were Android, and 4% were macOS. The dataset used for the experiments comprises DNS queries for RDNS servers observed from January 2017 to February 2018 on the campus network, which total 372 GB in size. For blacklist L_B, we combined three public lists of malicious domains [29]–[31]; for whitelist L_W, we used the top one million domains provided by Alexa [32]. Details of the data sources are given in Table 1.

We set the parameters in our approach as follows: $t_\alpha = 90$ and $t_\beta = 1.0$. Also, the number of iterations, number of dimensions, and learning rate for Word2Vec were set to 250,000, 100, and 0.0005, respectively. These five

TABLE 3. Characteristics of multivariate mixture models.

Model	Distribution	Volume	Shape	Orientation
EII	spherical	equal	—	—
VII	spherical	variable	—	—
EEI	diagonal	equal	equal	—
VEI	diagonal	variable	equal	—
EVI	diagonal	equal	variable	—
VVI	diagonal	variable	variable	—
EEE	ellipsoidal	equal	equal	equal
EVE	ellipsoidal	equal	variable	equal
VEE	ellipsoidal	variable	equal	equal
VVE	ellipsoidal	variable	variable	equal
EEV	ellipsoidal	equal	equal	variable
VEV	ellipsoidal	variable	equal	variable
EVV	ellipsoidal	equal	variable	variable
VVV	ellipsoidal	variable	variable	variable

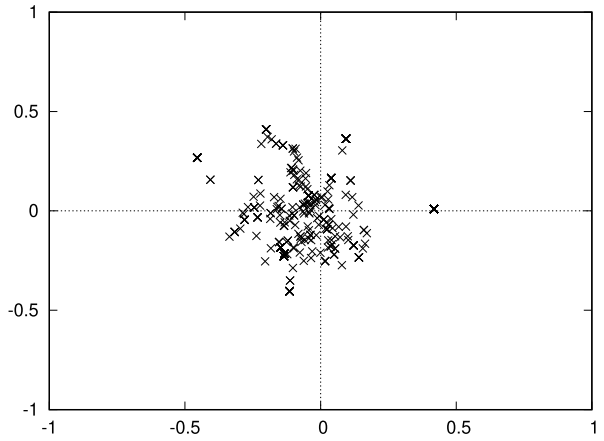
parameters were determined experimentally. Parameter optimization will be addressed in future work. In soft clustering, the BIC (bayesian information criterion) was used for model decision. The parameters were selected according to the BICs of 14 types of multivariate mixture models and up to 10 clusters. The characteristics of the types of multivariate mixture models are summarized in Table 3. For further details, refer to [28].

For comparison with the proposed approach, we implemented the two different approaches for classifying malicious queries described in [7] and [24]. The specifications of a machine for implementation are given in Table 2. The first implementation uses the similarity of character strings in domain names, and the second implementation uses Doc2Vec, which is a well-known extension to Word2Vec. While Word2Vec derives the feature vectors of words, Doc2Vec derives the feature vectors of documents. We set the maximum distance between words for measuring co-occurrences to 5 for Doc2Vec; the other parameters were set to the same values as those used for the proposed approach.

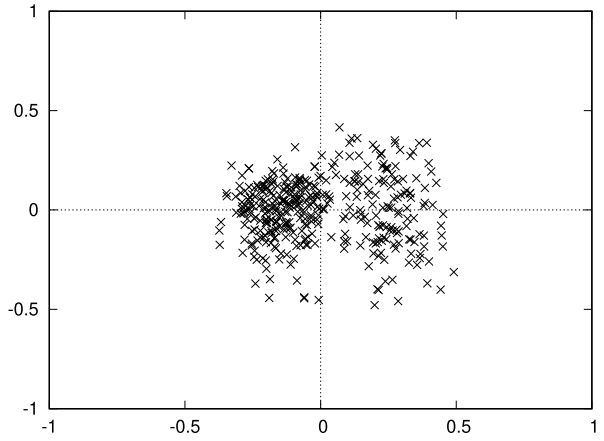
B. CLASSIFICATION ACCURACY

The authors of [33], [34] reported that many types of malware communicate through TXT-type queries. Therefore, we considered TXT-type queries in which the domain names matched blacklist entries to be malicious. Based on this criterion, 388 queries with 158 unique domains were detected from the dataset.

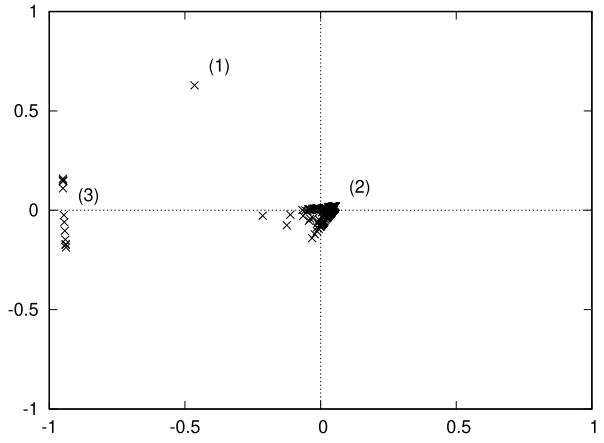
Figure 4 shows the experimental classification results, using multidimensional scaling to visualize the similarity among malicious queries. In the figure, each symbol represents a malicious query, and the distance between symbols indicates the similarity between the malicious queries. Because the symbols in Figures 4(a) and 4(b) are scattered, it is difficult to determine the similarity of the malicious queries. The respective reasons for performance deterioration in the two compared implementations are (1) the limitations



(a) Berger et al. [7]



(b) Le et al. [24]



(c) Our work

FIGURE 4. Experimental classification results.

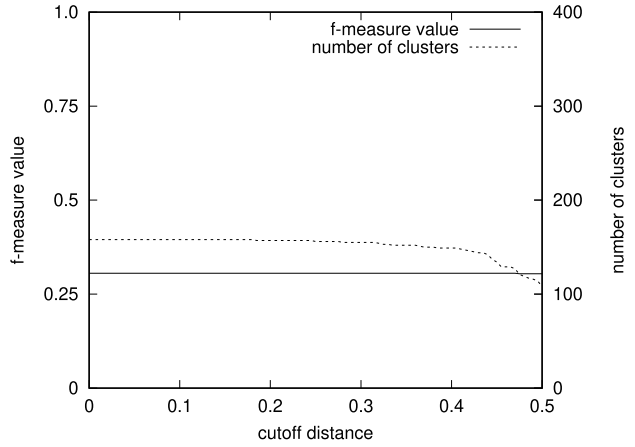
of classifying malicious queries based on the superficial similarity of character strings in domain names, and (2) the influence of surrounding queries that are unnecessary for estimating their causes. In contrast, our approach clearly classifies the 388 malicious queries detected through blacklists into 3 clusters, suggesting the possibility for efficient analysis.

The numbers of malicious queries classified into clusters (1), (2), and (3) shown in Figure 4(c) were 1, 375, and 12, respectively. We pursued their causes by investigating both the malicious queries themselves and the surrounding queries based on several services, such as web search, WHOIS [35], and domain reputation [36]–[39]. Only one malicious query was classified into cluster (1) because so few surrounding queries occurred either before or after the malicious query; namely, only seven queries to two domains were observed during the period. In such a case, i.e., when there are too few surrounding queries, it is difficult for the proposed approach to correctly derive similarity. In cluster (2), queries related to domain reputation frequently occurred before and after the malicious queries, for example, queries to `spamhaus.org`, `abuseat.org`, and `barracudacentral.org`. Accordingly, we believe that the malicious queries in cluster (2) were caused by misdetection of communications from some security appliances. In cluster (3), queries to BitTorrent tracking sites occurred before and after the malicious queries, for example, to `opentrackr.org`, `asnet.pw`, and `blackunicorn.xyz`. The communications were to domains included in the blacklists, and several studies have reported malware that use P2P for interactions [40], [41]. Accordingly, we attribute the malicious queries in cluster (3) to be due to malware infection. The results confirmed that each cluster consists of malicious queries with a common cause, which suggests the possibility for accurate classification.

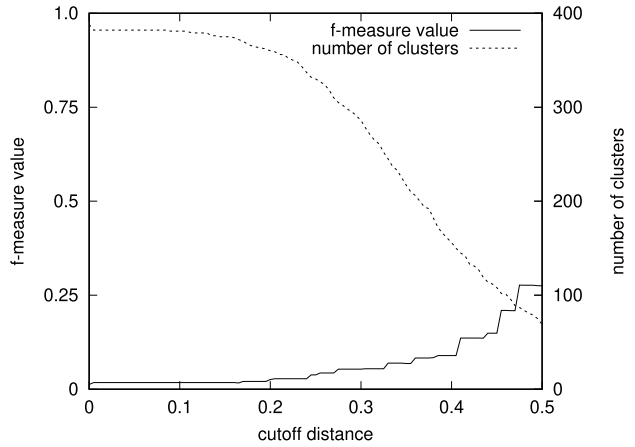
Figure 5 shows the relationships between F-measure value, number of clusters and cutoff distance in clustering malicious queries, where F-measure value is a common metric for quantitatively evaluating classification accuracy [42]. This metric penalizes the noise in each cluster, as follows:

$$\sum_{A\in\mathbb{A}} \frac{|A|}{N} \max_{C\in\mathbb{C}} \frac{2\mathcal{P}(A,C)\mathcal{R}(A,C)}{\mathcal{P}(A,C)+\mathcal{R}(A,C)}.$$

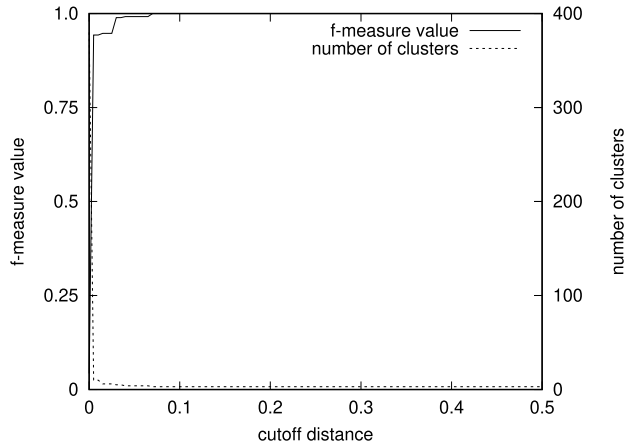
Here, $\mathcal{P}(A,C)$ and $\mathcal{R}(A,C)$ are given by $|A\cap C|/|A|$ and $|C\cap A|/|C|$; $\mathbb{C}$ is a set of clusters, each of which consists of malicious queries within a cutoff distance, and $\mathbb{A}$ is a set of true clusters revealed by the above analysis; N indicates the total number of elements, and $|\cdot|$ indicates the number of set elements. We used UPGMA (unweighted pair-group method using arithmetic averages) as a hierarchical clustering algorithm [43]. The solid line in the figure indicates the relationship between F-measure value and cutoff distance, whereas the dashed line indicates the relationship between number of clusters and cutoff distance. A cutoff distance ranges from 0 to 1.0, where 0 means that queries with the same similarity are classified into each cluster, and 1.0 means that all queries are classified into one cluster. In Figures 5(a) and 5(b), the highest F-measure value was roughly 0.25 at the cutoff distance of 0.5, at which point the number of clusters was over 80. In Figure 5(c), the F-measure value and number of clusters reached the maximum and minimum at the cutoff distance of 0.07, and after that, the two values remained



(a) Berger et al. [7]



(b) Le et al. [24]



(c) Our work

FIGURE 5. F-measure value and number of clusters with respect to cutoff distance in clustering malicious queries.

unchanged with increasing the cutoff distance up to 0.5. The results indicate that the proposed approach arranges similar malicious queries closely and different malicious queries far away. Thus this approach enables classification with a high accuracy compared to the two implementations.

TABLE 4. Number of queries for each cause in three types of data.

	Cause (1)	Cause (2)	Cause (3)
(A)	1	12	12
(B)	1	1	12
(C)	0	375	0

Summarizing the evaluation results, our approach successfully classified the 388 malicious queries detected through blacklists into 3 clusters, each consisting of malicious queries with a common cause. These results indicate that administrators can pursue all the causes by investigating only representative queries of each cluster, and thereby swiftly address the problem of infected machines in the network.

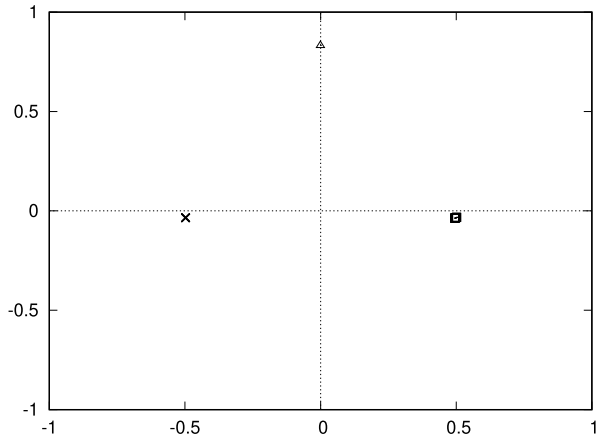
C. CLASSIFICATION ACCURACY FOR QUERY DATA WITH VARIOUS DEVIATIONS

For evaluation as mentioned in Section IV-B, we used 388 queries with 158 unique domains that ware classified into 3 causes. The data of these queries had large deviations because one cluster consisted of 375 queries with a common cause. Accordingly, we evaluated our approach using three types of query data to clarify the impact of data deviations on classification accuracy. Table 4 gives the number of queries for each cause in three types of data generated by randomly removing these 388 queries, where the cause number corresponds to the cluster number in Figure 4(c).

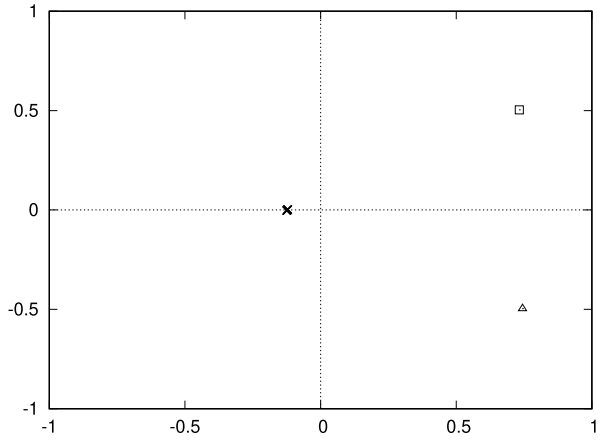
Figure 6 shows the experimental classification results for three types of query data, using multidimensional scaling to visualize the similarity among malicious queries. Each symbol represents a malicious query, each symbol type represents the cause of a malicious query, and the distance between symbols indicates the similarity between the malicious queries. In Figures 6(a), 6(b), and 6(c), the malicious queries with the same cause were classified into the same cluster; the F-measure values of query data (A), (B), and (C) were maximized at the cutoff distances of 0.005, 0.005, and 0.03, respectively. The results indicate that the proposed approach arranges similar malicious queries closely and different malicious queries far away for all the query data. Thus, the classification accuracy of this approach is unaffected by data deviations.

D. CALCULATION TIME

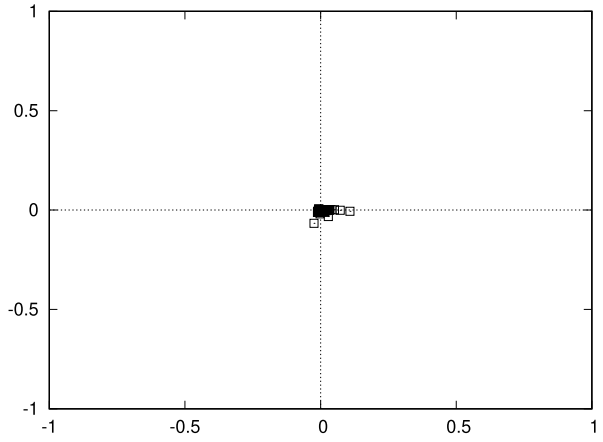
The calculation time for each approach is listed in Table 5. Note that the time required to select the query sub-log is excluded, since this can be processed in advance. The times for the two other implementations are 1.505 s and 1418.474 s, whereas our proposed approach had the worst time at 3976.791 s. The portion requiring the most computation time in the proposed approach is the processing of Word2Vec and soft clustering in the numerical representation function, which accounted for 83% of the total time. Based on this result, we investigated ways of improving the proposed approach.



(a) Query data (A)



(b) Query data (B)



(c) Query data (C)

FIGURE 6. Experimental classification results for three type of query data. (a) Query data (A). (b) Query data (B). (c) Query data (C).

Figure 7 shows the relationships between calculation time, loss value, and number of iterations in Word2Vec, where loss value is the sum of deviations between the predicted value and correct value when deriving co-occurrences. The solid line in the figure indicates the relationship between calculation time and number of iterations, whereas the dashed line indicates

TABLE 5. Experimental calculation time.

Berger et al. [7]	Le et al. [24]	Our work
1.505 s	1418.474 s	3976.791 s

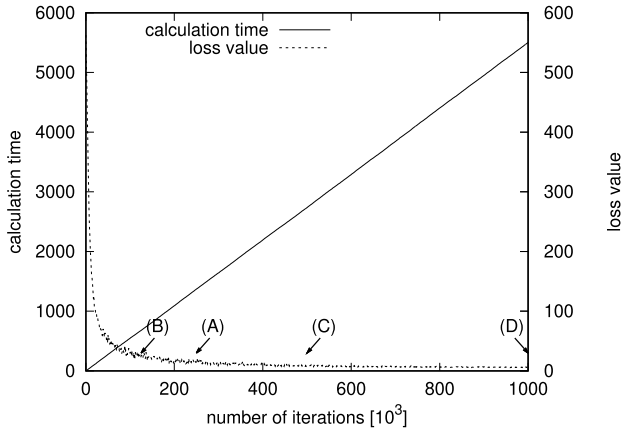


FIGURE 7. Calculation time and loss value with respect to number of iterations in Word2Vec .

the relationship between loss value and number of iterations. From the solid line increasing linearly we can infer the importance of keeping the number of iterations low in order to shorten calculation time. The point indicated by (A) is where the number of iterations is 250,000, which was used in the previous experiments, whereas (B), (C), and (D) respectively indicate 125,000, 500,000, and 1,000,000 iterations. The loss values at (A), (B), (C), and (D) are respectively 12.844, 25.892, 8.585, and 5.876, indicating that the loss values have nearly converged at point (B). Moreover, we examined the classification results for points (B), (C), and (D) and found no difference between them and the classification results for point (A), results that are similar to those in Figure 4(c). This is because the slight differences in co-occurrences are absorbed in the subsequent soft clustering step. It is clear from the results that calculation time can be significantly shortened by terminating iteration on the basis of loss value convergence.

In the soft clustering using 14 types of multivariate mixture models with the number of clusters varying from 1 to 20, the combination with the lowest BIC value, i.e., the combination best fitting the data, was EII with 4 clusters. We examined the calculation times of the processes and found that most of the time was taken up by complex models dealing with multiple variables, such as EEV, VEV, EVV, and VVV. Figure 8 shows the relationships between calculation time, BIC value, and number of clusters for the EEV model. Note that in the figure, calculation times and BIC values are shown relative to those of the EII model. We focused on EEV because it showed the lowest BIC among the complex models. Compared with the EII model, it can be seen that the EEV model requires a great deal of calculation time as the number of clusters increases to fit the data. The relative calculation time for 5 clusters was about 2.7, for 10 clusters was about 12,

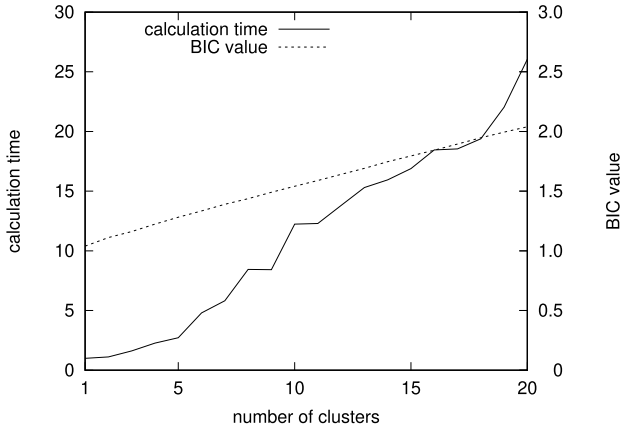


FIGURE 8. Calculation time and BIC value with respect to number of clusters in EEV model.

TABLE 6. Qualitative comparison of two other implementations and our approach.

	Berger et al. [7]	Le et al. [24]	Our work
classification accuracy	poor	poor	good
calculation time	good	fair	poor
efficiency for analysis	poor	poor	good
flexibility of approach	poor	fair	good
versatility of approach	poor	fair	good

and for 20 clusters was about 26. In addition, BIC is kept at a constantly high value, demonstrating a bad fit with the data. Specifically, the relative BIC value for 5 clusters was about 1.2, whereas for 10 clusters, it was about 1.5. It is clear from this result that the calculation time can be significantly shortened by successively excluding incompatible models based on the BIC value.

By improving our approach based on the above-mentioned two results, we confirmed that the calculation time can be reduced from 3976.791 s to 1442.302 s. The calculation time of this improved approach is almost equal to that of the method in [24].

E. QUALITATIVE COMPARISON

A qualitative comparison of the approaches is presented in Table 6. The factors for comparison are classification accuracy, calculation time, efficiency of analysis, flexibility, and versatility of approach. Looking at the results, firstly, the proposed approach has a higher classification accuracy than the other approaches, as discussed in Section IV-B. Since the investigation scope is limited to only representative queries in the classification results, the load on the administrator can be substantially reduced. On the other hand, the proposed approach requires a large computation time, although this can be improved as discussed in Section IV-D. Secondly, existing approaches based on the similarity of domain name character strings simply classify communications with the same domain as identical malicious queries, whereas our proposed approach focuses on both malicious queries and the accompanying queries, enabling an extremely flexible classification. For example, the approach can (1) distinguish malware communications to the same domain as different malicious queries on the basis of the success or failure of communication, and (2) distinguish communications consisting of multiple domains in coordination, as typified by malvertising [44] and malware distribution network [45], as the same malicious query on the basis of similarity of the destination. Finally, the proposed approach is highly versatile, having a range of applications not limited to DNS query logs. More concretely, it can potentially be applied to various system logs, such as those for anomaly detection, firewall, and web proxy.

One of the most essential tasks keeping the detection performance is to update blacklists. The proposed approach is based on the following observations: the surrounding queries that occur with a malicious query detected through blacklists depend on the cause of the malicious query, and accordingly, when malicious queries detected before and after updating involve similar surrounding queries, they will be classified into the same cluster. Thus, the classification accuracy of this approach is unaffected by updating blacklists.

An important issue is poisoning attacks against the training model in our approach. A poisoning attack pollutes the query log by initiating queries that looks like normal name resolution with spoofed IPs, and thereby might violate the soft clustering with gaussian mixture models to the distributed representation of queries. Adversarial machine learning [46] including such an attack is one of the interesting research fields. For example, Suciu *et al.* [47] proposed FAIL models that accurately evaluate the threat of realistic attacks against machine learning systems for improving defense mechanisms. Jagielski *et al.* [48] designed a new principled defense algorithm with significantly increased robustness against a large class of poisoning attacks. In addition to the findings reported in [47], [48], we expect that detection methods based on DNS traffic behaviors [49] are effective for this type of attack. This is because the attack will indicate similar behaviors to Kaminsky attacks by repeatedly initiating queries with spoofed IPs.

As mentioned in Section IV-B, the accuracy of the classification results using the proposed approach is determined based on our analysis. There are two main reasons for this. First, unlike datasets of other formats such as traffic [50], spam [51], malware binaries [52], and C&C domains [53], the format of query logs does not have a public dataset. Several studies have surely focused on collecting DNS datasets. For example, Kountouras *et al.* [54] implemented a system, Thales, that creates massive amounts of malicious domain names by distilling freely available and multiple sources. Pearce *et al.* [55] developed a scalable, accurate, and ethical system, Iris, that measures global name resolution with active manipulation for tracking the trends of domain names that evolve over time. However, various activities are reflected by query logs observed on private networks, and accordingly, DNS datasets including the query logs still have privacy concerns. Second, there are no established methods to evaluate the classification results because there are no studies that have

the same focus and approach as this one. Although there is some lack of objectivity, we believe that the results of our research hold important implications for subsequent studies with respect to guidelines and standards.

V. CONCLUSION

In this study, we aimed to classify malicious DNS queries detected through blacklists by their causes. Unlike the conventional classification approach, which is based on the superficial similarity of character strings and hierarchical structures in domain names, our cause-based classification drastically reduces the number of malicious DNS queries to be investigated because the investigation scope is limited to only representative queries in the classification results. Through experiments, we confirmed that our approach could group the 388 malicious queries into 3 clusters, each comprising queries with a common cause. These results indicated that administrators can briefly pursue all the causes by investigating only representative queries of each cluster. By enabling administrators to swiftly address the problem of infected machines in the network, our approach is able to dramatically improve network security.

The most important contribution of this study is that it showed that surrounding queries help in classifying malicious queries by their causes. The results can potentially be applied not only to DNS query logs but also to various system logs, such as those for anomaly detection, firewall, and web proxy. Moreover, we believe that our findings enhance both the detection of various malicious activities and their precise classification, with cross-sectional analysis of multiple system logs through the SIEM (security information and event management) [56].

In future work, we plan to evaluate the proposed approach by using multiple system logs through the SIEM. We will also consider adding a new function to remove clusters unrelated to malware communications.

REFERENCES

[1] J. A. Lewis. (2018). *Economic Impact of Cybercrime No Slowing Down*. [Online]. Available: https://www.csis.org/analysis/economic-impact-cybercrime
[2] B. Sun, M. Akiyama, T. Yagi, M. Hatada, and T. Mori, "Automating URL blacklist generation with similarity search approach," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 4, pp. 873–882, 2016.
[3] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Efficient and accurate behavior-based tracking of Malware-control domains in large ISP networks," *ACM Trans. Privacy Secur.*, vol. 19, no. 2, p. 4, 2016.
[4] N. Kheir, F. TranPierre, and C. Deschamps, "Mentor: Positive DNS reputation to skim-off benign domains in Botnet C&C blacklists," in *Proc. IFIP Int. Inf. Secur. Conf.*, 2014, pp. 1–14.
[5] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of Malware blacklists," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2014, pp. 1–21.
[6] R. Romero-Gomez, Y. Nadji, and M. Antonakakis, "Towards designing effective visualizations for DNS-based network threat analysis," in *Proc. IEEE Symp. Vis. Cyber Secur. (VizSec)*, Oct. 2017, pp. 1–8.
[7] 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.
[8] D. Plonka, and P. Barford, "Context-aware clustering of DNS query traffic," in *Proc. SIGCOMM Conf. Internet Meas.*, 2008, pp. 217–230.
[9] K. Lasota and A. Kozakiewicz, "Analysis of the similarities in malicious DNS domain names," in *Proc. FTRA Int. Conf. Secure Trust Comput., Data Manage. Appl.*, 2011, pp. 1–6.
[10] P. Zhang, T. Liu, Y. Zhang, J. Ya, J. Shi, and Y. Wang, "Domain Watcher: Detecting malicious domains based on local and global textual features," *Proc. Comput. Sci.*, vol. 108, pp. 2408–2412, 2017.
[11] 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.
[12] S. Špacek, M. Laštovickam, M. Horák, and T. Plesník, "Current issues of malicious domains blocking," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 551–556.
[13] S. Špacek, M. Laštovickam, M. Horák, and T. Plesník, "DNS firewall data visualization," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 743–744.
[14] P. Vixie. (2018). *DNS Response Policy Zones (RPZ)*. [Online]. Available: https://tools.ietf.org/html/draft-vixie-dnsop-dns-rpz-00
[15] 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. Secur.*, vol. 16, no. 4, p. 14, 2014.
[16] J. Li, X. Ma, L. Guodong, X. Luo, J. Zhang, W. Li, and X. Guan, "Can we learn what people are doing from raw DNS queries?" in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 2240–2248.
[17] H. Cui, J. Yang, Y. Liu, Z. Zheng, and K. Wu, "Data mining-based DNS log analysis," *Ann. Data Sci.*, vol. 1, nos. 3–4, pp. 311–323, 2014.
[18] P. Robberechts, M. Bosteels, J. Davis, and W. Meert, "Query log analysis: Detecting anomalies in DNS traffic at a TLD resolver," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2018, pp. 55–67.
[19] 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.
[20] 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.
[21] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," *IEEE Access*, vol. 3, pp. 1132–1142, 2015.
[22] X. Luo, L. Wang, Z. Xu, K. Chen, J. Yang, and T. Tian, "A large scale analysis of DNS water torture attack," in *Proc. Int. Conf. Comput. Sci. Artif. Intell.*, 2018, pp. 168–173.
[23] 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. Secur.*, vol. 64, pp. 1–15, Jan. 2017.
[24] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.
[25] A. Satoh, Y. Nakamura, D. Nobayashi, K. Sasai, G. Kitagata, and T. Ikenaga, "Clustering malicious DNS queries for blacklist-based detection," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 7, pp. 1404–1407, 2019.
[26] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," 2017, *arXiv:1701.07179*. [Online]. Available: https://arxiv.org/abs/1701.07179
[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[28] L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery, "Mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models," *R J.*, vol. 8, no. 1, pp. 289–317, 2016.
[29] *DNS-BH*. Accessed: Mar. 1, 2018. [Online]. Available: https://www.malwaredomains.com
[30] *HpHosts*. Accessed: Mar. 1, 2018. [Online]. Available: https://hosts-file.net
[31] *Abuse.Ch*. Accessed: Mar. 1, 2018. [Online]. Available: https://abuse.ch
[32] *Alexa*. Accessed: Mar. 1, 2018. [Online]. Available: http://www.alexa.com
[33] H. Ichise, Y. Jin, and K. Iida, "Analysis of DNS TXT record usage and consideration of Botnet communication detection," *IEICE Trans. Commun.*, vol. 101, no. 1, pp. 70–79, 2018.
[34] C. J. Dietrich, C. Rossow, F. C. Freiling, H. Bos, M. van Steen, and N. Pohlmann, "On Botnets that use DNS for command and control," in *Proc. 7th Eur. Conf. Comput. Netw. Defense*, Sep. 2011, pp. 9–16.
[35] *WHOIS Search, Domain Name, Website, and IP Tools*. Accessed: Sep. 10, 2018. [Online]. Available: https://who.is
[36] *VirusTotal*. Accessed: Sep. 10, 2018. [Online]. Available: https://www.virustotal.com

[37] *Web of Trust*. Accessed: Sep. 10, 2018. [Online]. Available: https://www.mywot.com
[38] *SimilarWeb*. Accessed: Sep. 10, 2018. [Online]. Available: https://www.similarweb.com
[39] *NetValuator*. Accessed: Sep. 10, 2018. [Online]. Available: http://www.netvaluator.com
[40] R. Cuevas, M. Kryczka, R. González, A. Cuevas, and A. Azcorra, "TorrentGuard: Stopping scam and malware distribution in the BitTorrent ecosystem," *Comput. Netw.*, vol. 59, pp. 77–90, Feb. 2014.
[41] A. D. Berns, "Searching for Malware in BitTorrent," Dept. Comput. Sci., Univ. Iowa, Iowa, IA, USA, Tech. Rep. UICS-08-05, 2008, pp. 1–10.
[42] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Inf. Retr.*, vol. 12, no. 4, pp. 461–486, 2009.
[43] D. Müllner, "Fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python," *J. Stat. Softw.*, vol. 53, no. 9, pp. 1–18, 2013.
[44] C. Dwyer and A. Kanguri, "Malvertising—A rising threat to the online ecosystem," *J. Inf. Syst. Appl. Res.*, vol. 10, no. 3, pp. 29–37, 2017.
[45] Z. Behfarshad, "Survey of malware distribution networks," Dept. Electr. Comput. Eng., Univ. Brit. Columbia, Vancouver, BC, Canada, Tech. Rep. 2, 2012, pp. 1–13.
[46] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," in *Proc. SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 2154–2156.
[47] O. Suciu, R. Marginean, Y. Kaya, H. Daume, and T. Dumitras, "When does machine learning FAIL? generalized transferability for evasion and poisoning attacks," in *Proc. USENIX Conf. Secur. Symp.*, 2018, pp. 1299–1316.
[48] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 19–35.
[49] S. Torabi, A. Boukhtouta, C. Assi, and M. Debbabi, "Detecting Internet abuse by analyzing passive DNS traffic: A survey of implemented systems," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3389–3415, 4th Quart., 2018.
[50] *Traffic Data from Kyoto University's Honeypots*. Accessed: Jun. 15, 2019. [Online]. Available: http://www.takakura.com/kyoto_data/
[51] *UCI Machine Learning Repository: Spambase Data Set*. Accessed: Jun. 15, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase
[52] *MalShare*. Accessed: Jun. 15, 2019. [Online]. Available: https://malshare.com
[53] *DGArchive-Fraunhofer FKIE*. Accessed: Jun. 15, 2019. [Online]. Available: https://dgarchive.caad.fkie.fraunhofer.de
[54] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, R. Joffe, "Enabling network security through active DNS datasets," in *Proc. Int. Symp. Res. Attacks Intrusions Defenses*, pp. 188–208, 2016.
[55] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, P. N. Weaver, and V. Paxson, "Global measurement of DNS manipulation," in *Proc. USENIX Secur. Symp.*, 2017, pp. 307–323.
[56] O. Podzins and A. Romanovs, "Why SIEM is irreplaceable in a secure IT environment?" in *Proc. Open Conf. Elect. Electron. Inf. Sci. (eStream)*, Apr. 2019, pp. 1–5.

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

YUTAKA NAKAMURA received the D.E. degree in computer science from the Nara Institute of Science and Technology. He is currently a Professor with the Information Science Center, Kyushu Institute of Technology, Japan. His research interests include server management, the internet measurement, and network security. He is a member of IEICE and IPSJ.

YUTAKA FUKUDA (M'02) received the D.E. degree in computer science from the Kyushu Institute of Technology. He is currently an Assistant Professor with the Information Science Center, Kyushu Institute of Technology, Japan. His research interests include performance evaluation of computer networks, wireless networks, and transport protocols. He is a member of IEICE.

KAZUTO SASAI received the D.S. degree in earth and planetary science from Kobe University. He is currently an Associate Professor with the Graduate School of Science and Engineering, Ibaraki University, Japan. His research interests include network management, network analysis, complex systems, social systems, and agent-based systems. He is a member of IEICE, IPSJ, JSAI, and SICE.

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 IEICE and IPSJ.