IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 22, 2019, accepted September 6, 2019, date of publication September 10, 2019,
date of current version September 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940554

Malicious Domain Names Detection Algorithm Based on Lexical Analysis and Feature Quantification

HONG ZHAO[1], ZHAOBIN CHANG[1], WEIJIE WANG[1], AND XIANGYAN ZENG[2]
[1]School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
[2]Department of Mathematics and Computer Science, Fort Valley State University, Fort Valley, GA 31030, USA

Corresponding author: Zhaobin Chang (1510998508@qq.com)

This work was supported in part by the National Science Foundation of China under Grant 51668043 and Grant 61262016, in part by the CERNET Innovation Project under Grant NGII20160311 and Grant NGII20160112, and in part by the Gansu Science Foundation of China under Grant 18JR3RA156.

ABSTRACT Malicious domain names usually refer to a series of illegal activities, posing threats to people's privacy and property. Therefore, the problem of detecting malicious domain names has aroused widespread concerns. In this study, a malicious domain names detection algorithm based on lexical analysis and feature quantification is proposed. To achieve efficient and accurate detection, the method includes two phases. The first phase checks an observed domain name against a blacklist of known malicious uniform resource locator (URLs). The observed domain name is classified as being definitely malicious or potentially malicious based on its edit distances to the domain names on the blacklist. The second phase further evaluates a potential malicious domain name by its reputation value that represents its lexical feature and is calculated based on an N-gram model. The top 100,000 normal domain names in Alexa are used to obtain a whitelist substring set using the N-gram method in which each domain name excluding the top-level domain is segmented into substrings with the length of 3, 4, 5, 6 and 7. The weighted values of the substrings are calculated according to their occurrence counts in the whitelist substring set. A potential malicious domain name is segmented by the N-gram method and its reputation value is calculated based on the weighted values of its substrings. Finally, the potential malicious domain name is determined to be malicious or normal based on its reputation value. The effectiveness of the proposed detection method has been demonstrated by experiments on public available data.

INDEX TERMS Malicious domain names, N-gram, domain name substring, edit distance, reputation value.

I. INTRODUCTION

Malicious domain names are widely used by attackers for illegal activities in Domain Name System (DNS). As shown in some reports [1], [2]. The number of malicious domain names has grown to the point where they cannot be ignored. Hence, the detection of malicious domain names plays a major role in ensuring the network security.

DNS, a core component of the Internet that provides flexible decoupling of a service's domain name and the hosting IP addresses, has been widely used in network communications, e-business, and mess media [3]. Almost all Internet applications need to use DNS to resolve domain names and achieve resource location [4]. On the other hand, DNS services have been abused to perform various attacks. Malicious attackers use the defects of DNS, such as lacking of self-detection of malicious behavior, to attack Internet. Therefore the security of DNS is one of the key internet security challenges.

Through a recursive query, malicious attackers resolve normal DNS resolution requests to their malicious servers [5], [6]. In this process, malicious attackers apply domain-flux or fast-flux technique to locate their Command and Control (C&C) server by automatically generating a large number of non-existent domain names using domain generation algorithms (DGA) [7]–[11]. In order to contact the infected host, each malicious machine may use DGA to produce a list of candidate C&C domains. The infected host then attempts to resolve these domain names by sending

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Esposito.

domain name resolution request until it gets a successful answer from the malicious domain name reserved in advance by the malicious machine master. This malicious domain name attack strategy is an effective technique to achieve malicious purposes. These resolution requests and failure records of non-existent domain names are forwarded multiple times among DNS servers, which takes huge amount of bandwidth and processing resources with the aim of making DNS server unavailable to users. Meanwhile, it will seriously affect the execution of normal domain name resolution tasks. If these malicious domain names are not identified accurately in a timely manner, all Internet services relying on DNS servers will be down and the results will be catastrophic.

Therefore, accurate and timely detection of malicious domain name attacks is crucial for the normal operation of Internet. The contributions of this study are described as follows:

- A two-phase detection mechanism is proposed to achieve efficient and accurate detection. The lexical features of domain names are used in both phases.
- The first phase checks the observed domain name against a blacklist of known malicious URLs. The edit distance is adopted to detect malicious domain names on a blacklist quickly and reduce time overhead.
- The second phase further evaluate the domain names that cannot be determined in the first phase. For this purpose, the reputation value of a domain name is calculated based on a whitelist substring set and used to classify it as normal or malicious. The N-gram method is used to build the whitelist substring set of known normal domain names. The weighted values of common substrings are calculated from their occurrences in the whitelist substring set. The reputation value of a domain name is calculated using the weight value of its substrings.
- The two phases address the detection problem from a comprehensive by checking against the blacklist and whitelist.

The rest of this paper is organized as follows. The literature review is given in section II. The framework and the methodology are introduced in section III. The experimental results are discussed in section IV. Finally, the conclusion and future work are given in section V.

II. LITERATURE REVIEW

Prior work on malicious domains detection can be summarized as approaches based on domain name blacklist detection, domain name semantic analysis and domain name query behavior analysis.

A. DOMAIN NAME BLACKLIST DETECTION

The domain name blacklisting technology explicitly compares an observed domain name with the domain names on a blacklist of known malicious URLs and then makes decisions to allow or decline a user request. For example, Lasota and Kozakiewicz [12] proposed a malicious domain names detection algorithm by extracting and analyzing the similarity characteristics of malicious domain names on a blacklist. Kuhrer *et al.* [13] discussed the efficiency of different blacklists including 15 public malware blacklists and 4 private malware blacklists from anti-virus vendors. They identified the unregistered domain names in listings using DNS. Zhao *et al.* [14] proposed a fast malicious domain names detection algorithm that clustered the domain names based on their length attribute values and used the edit distance between each domain in each domain name group and the domain names on a blacklist to identify malicious domain names. Sato *et al.* [15] proposed a malicious domain names detection algorithm that used co-occurrence relation between DNS queries to detect the domain name requests sent by an infected host in real time. Their algorithm achieves malicious domain names detection through analyzing the characteristics of the group that requests the same set of hosts.

B. DOMAIN NAME SEMANTIC ANALYSIS

To distinguish between normal and malicious domain names, researchers have analyzed semantic features using classification techniques. For example, Altay *et al.* [16] proposed a context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection, which analyzed the domain name features such as the length attribute, and keyword frequency to identify malicious domain names. Huang *et al.* [17] proposed a malicious URL detection algorithm by dynamically mining patterns without pre-defined elements, which are not necessarily assembled using any pre-defined items, to capture malicious URLs generated algorithmically by malicious programs. Zouina and Outtaj [18] proposed a lightweight malicious domain names detection system using support vector machines and six URL features, namely, URL length, number of hyphens, number of dots, number of numeric characters, a discrete variable that corresponds to the presence of an IP address in the URL, and finally, the similarity index. Schiavoni *et al.* [19] proposed a DGA classifier for real-time detection using the linguistic features. The linguistic features of significant characters ratio and n-gram normality score were estimated using Alexa top one million dataset. The mahalanobis distance measures was used to calculate the distance of unknown domains. If a distance was too large, it was classified as DGA, otherwise as normal.

C. DOMAIN NAME QUERY BEHAVIOR ANALYSIS

In addition to identifying malicious domain names based on specific string features, group behaviors of malicious domain name requests can be used for malicious domain names detection [20]. For example, Yadav *et al.* [21] proposed an algorithm that detected algorithmically generated domain-flux attacks through DNS traffic analysis to detect botnets, addressing the domain fluxing mechanism employed by the botnets such as Conficker, Zeus and Torpig. Rahbarinia *et al.* [22] proposed a behavior-based technique to track malware-controlled domain names. Their algorithm extracted user behavior patterns from DNS query logs beyond

the bipartite host-domain graph. Bilge *et al.* [23] proposed an exposure system that first extracted 15-dimensional features of domain names and then used J48 decision tree for classification. Antonakakis *et al.* [24] proposed a technique to detect DGA without reverse engineering, where they found that bots from the same botnet (using the same DGA) would have similar Non-Existent Domain (NX-Domain) responses.

Among the above-mentioned approaches, the domain name blacklist detection methods have the advantage of low detection time overhead and high detection precision rate. However, this kind of detection method is unable to effectively detect the newly generated domain names, which leads to low detection accuracy rates, high false positive rates and high false negative rates. The detection methods of domain name semantic analysis have the advantage of high detection accuracy rate. However, this kind of detection method is based on domain name blacklist to design detection features, which limits the detection range. Although the detection methods based on analysis of query behaviors of domain names have wide applications and high detection accuracy rates, these methods require a long data collection period. It is difficult to obtain a large amount of resolution data from both the local domain name server and the root domain name server. Therefore, these methods have high detection time overhead.

To overcome these issues, a new detection method based on lexical analysis and feature quantification is proposed that uses a blacklist of known malicious domain names and a whitelist of known normal domain names in two separate phases. Checking an observed domain name against a blacklist has the advantage of low time overhead. Analyzing its similarity to normal domain names on a whitelist can improve the detection accuracy. In addition, unlike current detection methods that analyze the lexical composition and structure of the whole domain names, the new method divides a domain name into multiple substrings and analyzes the features of the substrings from a linguistic and lexical composition perspective.

III. PROPOSED METHODOLOGY

This section describes the details of our methodology.

A. OVERVIEW

Fig. 1 presents the architecture of malicious domain names detection algorithm based on lexical analysis and feature quantification, which consists of two components: construction of domain name whitelist substring set and detection of malicious domain names. To construct a whitelist substring set, the normal domain names with high access frequency, excluding the top-level domain names, are segmented into multiple substrings by the N-gram method, and the weight value of a substring is calculated based on its occurrence number in the domain name whitelist substring set. The main goal of this phase is to obtain the occurrence of common substrings that will be used in analyzing potential malicious domain names. Malicious domain names detection consists of two

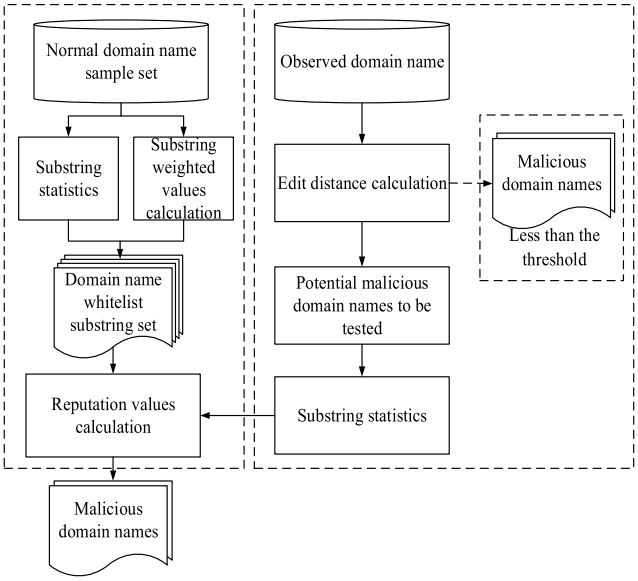

FIGURE 1. Flowchart of malicious domain names detection.

phases. The first phase classify an observed domain name as malicious or potential malicious. The observed domain name is identified as malicious if its edit distance to the domain names on the blacklist is less than a threshold value, otherwise it is considered to be potential malicious. The second phase further analyzes potential malicious domain names. A potential malicious domain name is segmented by the N-gram method. The reputation value of the potential malicious domain name is calculated based on the weighted values of its substrings and is used to determine the domain name is malicious or normal. A domain name is determined to be normal if its reputation value is greater than a threshold value.

B. CONSTRUCTION OF DOMAIN NAME WHITELIST SUBSTRING SET

To obtain the domain name whitelist substring set, we examined a large number of normal domain names in Alexa [25]. It is found that the domain name has a hierarchical structure.

Alexa rank is a list that Amazon measures the relative reputation of a domain name arranged by internet popularity [26], [27]. If a domain name ranks relatively high in the Alexa, it is more likely to be secure and normal [28].

***URL* (*Uniform Resource Locator*)**, a web address that is a reference to a web resource, is used to specify the location of the web resource on the network. The structure of URL is shown in Fig. 2. The URL is composed of several components such as protocol, path domain, top level domain, second level domain (SLD), and third level domain (TLD), etc. Top level domain, SLD, and TLD are together called as domain name [29]. The domain name is the name given to the real Internet IP address through the DNS, The top level domain is the domain name substring of the highest position in the domain name hierarchy architecture, including the national top-level domains (e.g., cn, us, and jp), and international top-level domains (e.g., com, net, and org) [30]. The SLD,

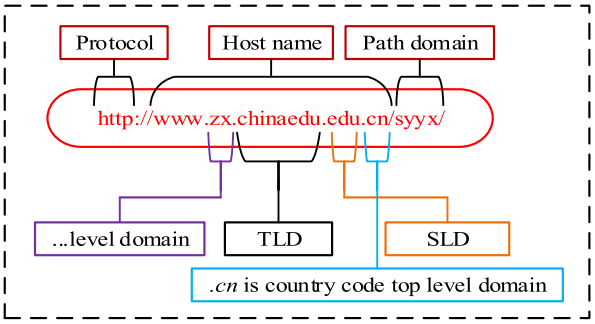

FIGURE 2. URL structure.

the most important part of a domain name, is located directly next to the top-level domain. The TLD is an ancillary domain given to the domain name and has various types depending on the services provided by the domain page. Given an example of domain name http://www. chinaedu.edu.cn, cn is China's top-level domain on the Internet; edu is the SLD, which represents education organization; chinaedu is the TLD, which represents China Educational Information Platform. Therefore, domain name substrings at each level have a specific meaning in its construction [31].

When the application process needs to map a host domain name to an IP address, the domain name resolution function is called, and the resolution function puts the converted domain name in the DNS request and sends it to the local domain name server via UDP message [32]. After the local domain name server searches the domain name, it returns the corresponding IP address in the reply message. At the same time, the domain name server must also have information connected to other servers to support forwarding that cannot be resolved. If the domain name server cannot answer the resolution request, the domain name server will become another customer in DNS. Then the resolution requests are forwarded to the DNS servers again where the top, second and other level domain names are located, until the query to the requested domain name.

It can be seen from the process of domain name resolution that the deeper level a domain name is at, the greater its forwarding number is, thus the heavier query load it creates to the system. On the contrary, the closer a domain name is to the top level domain, the smaller its forwarding number is, and thus the easier it can be found. Furthermore, because of the small quantity, short length and high popularity of top level domains, they are easily recognized. Therefore, malicious domain names are rarely found in the top-level domain, as normally exists in the secondary, tertiary and higher level-domains. Therefore, this study mainly focuses on other level domain substrings excluding the top level domain.

1) SUBSTRING STATISTICS

In this study, we use the N-gram model as described in [33] and [34]. The character string in the text is segmented by a sliding window with a size of N, and multiple contiguous sequence of length N are obtained, each of which is called a gram. For example, the 4-gram segmentation process for a character string *maliciousdomain* is shown in Fig. 3.

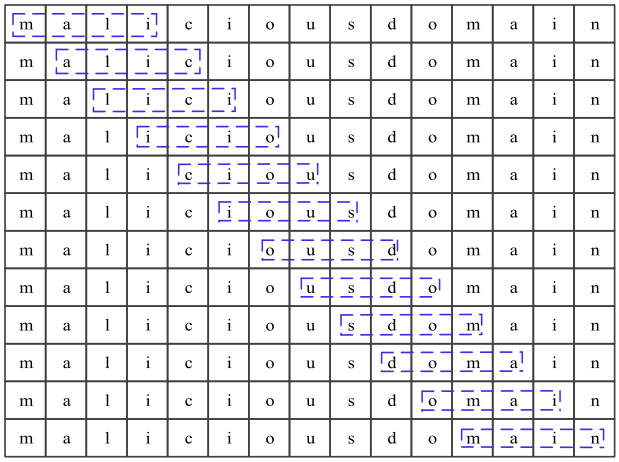

FIGURE 3. Process of 4-gram segmentation.

The N-Gram method is introduced to segment a given sequence of the text, the size of N will influence the number of gained domain name substrings. If the value of N is too small, the number of substrings obtained by segmentation will be large, which leads to high computational complexity and space complexity. If the value of N is too large, the number of substrings obtained by segmentation will be small, which leads to few character statistical feature information of URL [35]. Furthermore, the N-gram method has the ability to predict the occurrence of phenomena [36].

To determine appropriate N values, the top 100,000 domain names in Alexa are examined and the statistics of the lengths of other level domains is shown in Tab 1. It is noted that the length values in the [3, 7] interval is up to 97.63%. Therefore, the size of N is set to 3, 4, 5, 6 and 7, and each domain name excluding top level domain is segmented by the N-gram method to construct the whitelist substring set.

TABLE 1. Length proportion of other level domain excluding top level domain.

Length	≤ 2	3	4	5	6	7	≥ 8
Proportions(%)	0.55	5.39	20.09	29.13	29.21	13.81	1.82

An example of segmenting domain names by the n-gram method is shown in Fig. 4. After excluding the top level domain from groups.google.com, the SLD AND TLD are segmented by the N-gram method. The SLD substring sets are {goo, oog, ogl, gle, goog, oogl, ogle, googl, oogle, google}. And the TLD substring set is {gro, rou, oup, ups, grou, roup, oups, group, roups, groups}.

In order to the occurrence number of completely different domain name substrings, we select the Alexa's top 100,000 in this study, and each domain name excluding the top level domain is segmented into multiple substrings by the N-gram

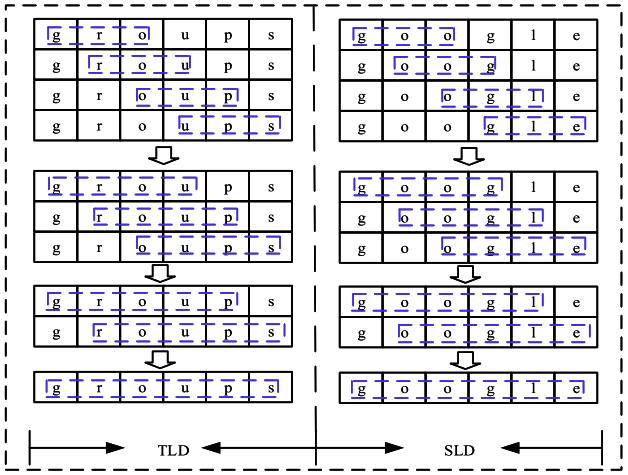

FIGURE 4. Principle diagram of domain name segmentation.

TABLE 2. Distribution of domain name substrings ($N = 3, 4, 5, 6, 7$).

N	Number of substrings
3	21,584
4	84,431
5	120,626
6	116,908
7	55,274
Total	398,823

method to construct the whitelist substring set. The occurrence number of substrings at each level domain is calculated by Eq. (1).

$$count(j) = L - N + 1 \tag{1}$$

where $count(j)$ $(j = 1, 2, \ldots, n)$ denotes the number of domain name substrings that are obtained from segmenting the j-th level domain of a domain name, L represents the length of j-th level domain, n denotes the maximum level number of a domain name, and $N\{N \in N * |3 \leq N \leq 7\}$ stands for the size of sliding window.

When the size of N is set to 3, 4, 5, 6 and 7, we gather statistics and analyze the distribution of the character in the domain name. The distribution of the completely different substrings is displayed in Tab. 2. Where the number of substrings with the N-sized of 3 is 21,584, with the N-sized of 4 is 84,431, with the N-sized of 5 is 120,626, with the N-sized of 6 is 116,908 and with the N-sized of 7 is 55,274, with a total of 398,823 substrings.

2) SUBSTRING WEIGHTED VALUES CALCULATION

Word frequency analysis is one of the most fundamental analytic methods in semantic analysis [37]. The frequency distribution of substrings is quite different between the normal domain names and the malicious domain names. To clearly illustrate the difference, we count the substring weight value of each domain name under different sliding window. The weight value of domain name substring can be calculated by Eq. (2).

$$W_{N-gram}(i) = log_2(\frac{C_{N-gram}(i)}{N}) \tag{2}$$

where W_{N-gram} $(N = 3, 4, 5, 6, 7)$ denotes the weight value of the i-th substring, $C_{N-gram(i)}$ stands for the total number of the occurrences of the i-th domain name substring after the top 100,000 domain names are segmented in Alexa.

398,823 substrings are extracted from the Alexa's top 100,000 domain names by the N-gram method, and each domain name substring weight value is calculated. According to these completely different domain name substrings, we construct the domain name whitelist substring set. We refer to these domain name substring weighted values to calculate the expected weight value for each observed domain name. Excerpt of some gram weights $(N = 3, 4, 5, 6, 7)$ in normal domain names from Alexa top 100,000 are shown in Tab. 3.

TABLE 3. Excerpt of some gram weights ($N = 3, 4, 5, 6, 7$) in normal domain names from Alexa top 100,000 domain names.

gram	$C_{N\text{-gram}}(i)$	$W_{N\text{-gram}}(i)$
ine	2510	9.708
ers	1544	9.007
nlin	1096	8.098
ster	568	7.149
irect	585	6.870
ogspo	294	5.614
hostin	161	4.745
vejour	125	4.380
rketing	167	4.576
olution	91	3.700

C. DETECTION OF MALICIOUS DOMAIN NAMES

1) IDENTIFYING POTENTIAL MALICIOUS DOMAIN NAMES

This subsection mainly includes domain name blacklist sample construction, edit distance calculation and difference degree value calculation.

a: CONSTRUCTION OF MALICIOUS DOMAIN NAME BLACKLIST

The domain name blacklist is used to determine whether an observed domain name is malicious.

b: EDIT DISTANCE CALCULATION

Edit distance (ED or Levenshtein Distance) [38] gives the minimum number of single-character operations (insertion, deletion, and substitution) required to convert one string into another. The computation of edit distance between two

domain names *str* and *str'* is a dynamic programming problem [39], and can be broken into a collection of sub-problems to calculate $lev(i, j)$ defined as follows:

$$lev(i,j) = \min \begin{cases} lev(i-1,j)+1 \\ lev(i,j-1)+1 \\ lev(i-1,j-1) + \begin{cases} 1, & if\ str(i) \neq str'(j) \\ 0, & if\ str(i) = str'(j) \end{cases} \end{cases} \tag{3}$$

where $i = 1, \ldots, |str|$, $j = 1, \ldots, |str'|$, $str(i)$ is the i-th character in *str* and *str'*(j) is the j-th character in *str'*. Assuming $lev(i, 0) = i$ and $lev(0, j) = j$. We start start with $i = 1$, $j = 1$ and alternately increment them by 1 each time until $i = |str|$, $j = |str'|$, the edit distance between an observed domain name *str* and a malicious domain name *str'* is defined as ED $(str, str') = lev(|str|, |str'|)$.

c: DIFFERENCE DEGREE VALUE CALCULATION

Difference degree value (DDV) [40] between domain name *str* and *str'* is defined as:

$$DDV = \frac{2ED(str, str')}{n+m} \tag{4}$$

where m and n are the length values of domain names *str* and *str'*, respectively. In Eq. (4), the DDV between domain name *str* and *str'* is proportional to the ED (*str*, *str'*), and inversely proportional to their lengths.

To determine if the observed domain name *str* is a malicious domain name or a potential malicious domain name, we compare DDV with a threshold $\lambda 1$, as shown below:

$$\begin{cases} \text{if DDV} < \lambda_1, & str \text{ is malicious} \\ \text{if } DDV \geq \lambda_1, & str \text{ } is \text{ potential malicious} \end{cases} \tag{5}$$

If the DDV between the observed domain name *str* and the domain name *str'*on the blacklist is less than the threshold λ_1, the observed domain name *str* is directly determined to be a malicious domain name. Otherwise it is necessary to further analyze the potential malicious domain name to finally determine whether it is a malicious domain name or not.

2) ANALYSIS OF POTENTIAL MALICIOUS DOMAIN NAMES

In this subsection, we introduce the process to make the final decision on a potential malicious domain name. As shown in Fig.5, a potential malicious domain names is first segmented by the N-gram method, and its reputation value is calculated according to the weighted values of its substrings in the whitelist substring set. The judgment of whether a potential malicious domain name is malicious is made based on the reputation value.

a: REPUTATION VALUE CALCULATION

The reputation value (RV) of a potential malicious domain name is calculated as the total weight values of its substrings

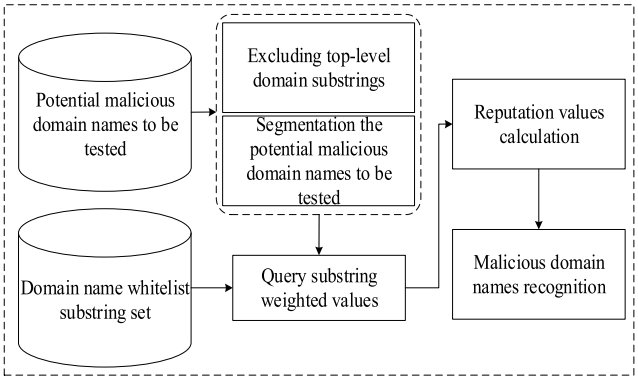

FIGURE 5. The framework of malicious domain names recognition.

in the whitelist substring set as shown below.

$$RV(l) = \sum_{i=1}^{m} W_{N-gram}(i) \tag{6}$$

where m is the total number of substrings of domain name l, $W_{N-\text{gram}}$ $(N = 3, 4, 5, 6, 7)$ represents the weight value of i-th substring which is referenced from 398,823 domain name substring weighted values (as shown in Tab. 3), l stands for a potential malicious domain name. Since the substrings of normal domain name appear more frequently in the whitelist substring set, the RV of normal domain name is larger. On the contrary, the substrings of malicious domain names appear less frequently in the whitelist substring set, the RV of malicious domain names is smaller. Therefore, we can threshold the RV value to distinguish between normal domain names and malicious domain names as below.

$$\begin{cases} \text{if RV}(l) < \lambda_2, & l \text{ is Malicious} \\ \text{if } RV(l) \geq \lambda_2, & l \text{ } is \text{ Normal domain name} \end{cases} \tag{7}$$

The threshold $\lambda 2$ is set based on the whitelist substring set.

D. EVALUATION CRITERIA

In this study, we use a confusion matrix [41], [42] to measure and evaluate the effectiveness of the proposed method in our experiments, as shown in Tab. 4. The following measure parameters are used to evaluate predictive performance of the proposed detection algorithm:

Accuracy Rate (AR) is the total number of correctly detected domain names divided by the total number of the detected domain names.

$$AR = \frac{N_{m\to m} + N_{n\to n}}{N_{m\to m} + N_{n\to n} + N_{m\to n} + N_{n\to m}} \tag{8}$$

Precision Rate (PR) is the number of the correctly predicted malicious domain names divided by the total number of the domain names that are predicted as malicious domain names.

$$PR = \frac{N_{m\to m}}{N_{m\to m} + N_{n\to m}} \tag{9}$$

TABLE 4. Confusion matrix parameters.

Actual	Predicted		
	Negative	Positive	Total
Negative	$N_{n\to n}$	$N_{n\to m}$	$N_{n\to n}+N_{n\to m}$
Positive	$N_{m\to n}$	$N_{m\to m}$	$N_{m\to n}+N_{m\to m}$
Total	$N_{n\to n}+N_{m\to n}$	$N_{n\to m}+N_{m\to m}$	Neg + Pos

TABLE 5. Experimental environment.

Parameters	Value
CPU	AMD A12-9700 2.5GHZ
GPU	AMD R8 M435DX
Memory	8GB
OS	64-bit Windows10
Platform	Jupyter Notebook
Python	3.5

False Negative Rate (FNR) is the number of the incorrectly predicted normal domain names divided by the total number of the malicious domain names.

$$FNR = \frac{N_{m\to n}}{N_{m\to n} + N_{m\to m}} \tag{10}$$

False Positive Rate (FPR) is the number of the incorrectly predicted malicious domain names divided by the total number of the normal domain names.

$$FPR = \frac{N_{n\to m}}{N_{n\to m} + N_{n\to n}} \tag{11}$$

where $N_n \to_n$ denotes the number of normal domain names that are correctly predicted as normal domain names, $N_m \to_n$ denotes the number of malicious domain names that are incorrectly predicted as normal domain names, $N_n \to_m$ denotes the number of normal domain names that are incorrectly predicted as malicious domain names, $N_m \to_m$ denotes the number of malicious domain names that are correctly predicted as malicious domain names.

IV. EXPERIMENTAL AND RESULT ANALYSIS

In order to evaluate the effectiveness of our proposed detection method. Here, we first introduce our experimental environment. Then, we introduce our datasets. Thereafter, we describe the experimental results in detail. Finally, we describe the performance result analysis and discussion.

A. EXPERIMENTAL ENVIRONMENT

The experimental environment is presented in Tab. 5.

B. DATA COLLECTION

Domain names in the whitelist and the blacklist mainly come from the public available data. The whitelist contains the top 100,000 domain names in Alexa list. Furthermore, each domain name excluding the top level domain is segmented into multiple substrings according to its domain level with the length of 3, 4, 5, 6 and 7 by the N-gram method. 398,823 completely different substrings are chosen as the domain name whitelist substring set.

The malicious domain names on the blacklist are collected from malwaredomains.com, malicious domain list, Zeus Tracker, Conficker, Torping, Symmni [43]-[48], etc.

In this study, 13,000 normal domain names from the Alexa and Anquan Organization [49], and 11,000 malicious domain names that are generated by the DGA from malicious domain list, the Phishing Tank [50], Newgoz and Shiotob [51], [52] are used as the test data in the experiment.

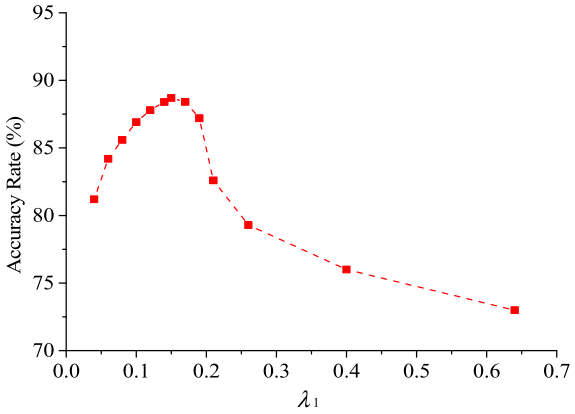

FIGURE 6. Accuracy rate of malicious domain names detected by the blacklist under different thresholds.

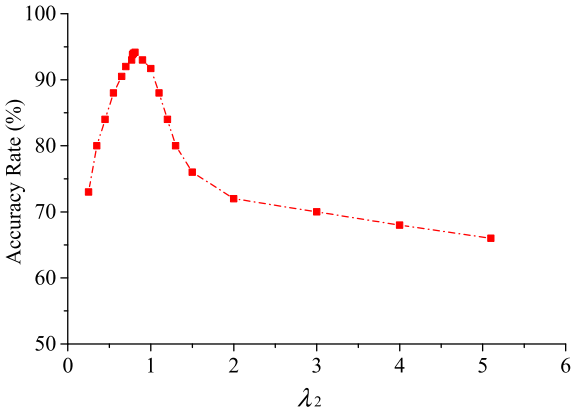

FIGURE 7. Accuracy rate of the 2-phase detection with different threshold λ2.

C. THRESHOLD SELECTION

The performance of the proposed method depends on the threshold parameters λ1 and λ2. Fig. 6 shows the accuracy rate of malicious domain names detected by the blacklist in the first phase. When the threshold λ1 is 0.16, the accuracy rate reaches an optimal level of 88.9%. Thus, in the following discussion, the threshold λ1 is set to 0.16.

Fig. 7 shows the detection accuracy rate of potential malicious domain names by the reputation value using the whitelist substring set. We can see that when the threshold

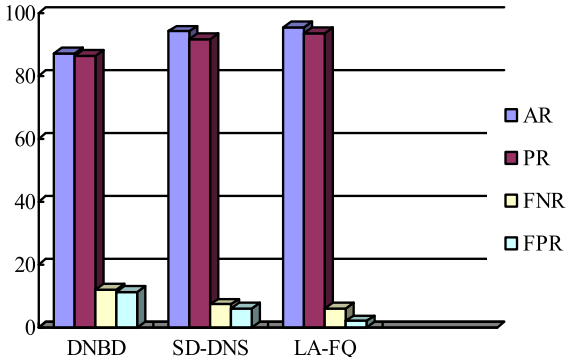

FIGURE 8. Performance comparison.

TABLE 6. Comparison of LA-FQ and other methods.

Method	AR(%)	PR(%)	FNR(%)	FPR(%)
H. Lin *et al.* [11]	93.16	91.04	6.47	5.13
D. Huang *et al.* [17]	**94.31**	93.35	7.01	**4.39**
Q. Hai *et al.* [20]	94.08	**93.50**	5.67	5.58
LA-FQ	94.16	93.33	**5.35**	4.91

is $\lambda 2 = 0.81$, and the detection accuracy rate reaches an optimum level of 94.58%.

D. EXPERIMENTAL RESULTS

The effectiveness of the proposed method is verified in this section. First, we demonstrate that combining blacklist and whitelist outperforms individual ones. The experiment was conducted using the popular domain name blacklist detection (DNBD), the statistical detection of domain name substring (SD-DNS) and the proposed lexical analysis and feature quantification (LA-FQ) with the same experimental conditions. Accuracy rate, precision rate, false positive rate and false negative rate are the measures for the effectiveness of the algorithms. Performance comparisons in terms of AR, PR, FNR and FPR are illustrated in Fig. 8.

Because the proposed method combines the information from the blacklist and the whitelist, the performance of LA-FQ outperforms DNBD and SD-DNS in all the measure. The detection accuracy rate of 94.16% and a precision rate of 93.33%. In addition, false negative rate and false positive rate are marginally decrease. The main reason is that the detection process of our study relies on the double threshold detection, which are obtained from multiple aspects and have more information than the features from a single aspect.

Tab. 6 illustrates the four metrics of LA-FQ and other methods (Lin *et al.* [11], D. Huang *et al.* [17], and Q. Hai *et al.* [20]) based on the evaluation criteria in this study. In order to facilitate comparison, we calculate the four metrics based on our experiments results. Huang *et al.* has highest AR than LA-FQ, but LA-FQ achieves the lowest FNR and running time (see Fig. 9).

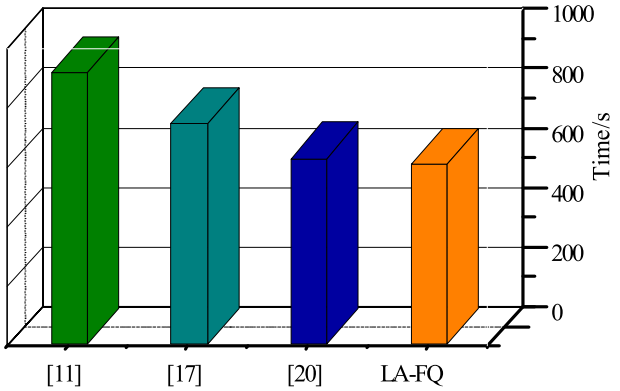

FIGURE 9. Running time of LA-FQ and other methods (H. Lin *et al.* [11], D. Huang *et al.* [17] and Q. Hai *et al.* [20]).

Although misjudging a normal domain name as a malicious domain name may instill inconvenience and trust issues to the operators of the website, the main work of this study is to detect malicious domain names accurately and reduce the false negative rate. Our proposed method has the best performance in this regard. Furthermore, our method is much easier to add new data when they become available. While the machine learning algorithms require a new training process of all the data, our approach only needs modifications to the threshold.

V. CONCLUSION AND FUTURE WORK

In this study, we propose a novel method based on lexical analysis and feature quantification for malicious domain names detection and compare them with two real-life malicious domain names detection models of DNBD and SD-DNS. Experimental results show that our approach not only performance in the efficiency and accuracy rate, but also the stronger generalization ability. It has a good practical value in defending against the Botnet, Spam and remote access Trojan attack, and can help security experts and organizations in their fight against cyber-crime.

Our goal is to detect these malicious domain names as early and accurately as possible, and help to prevent other users from falling victim of the same threats. However, our proposed method of using lexical features and characters distribution is not comprehensive and cannot detect all malicious domain names on the Internet, but if the malicious domain names are generated by randomly, our approach can detect them efficiently. Future work will be based on both further refinement of the methods, and a more sophisticated analysis using more substantial data sets.

REFERENCES

[1] *National Internet Emergency Center*. Accessed: Jan. 18, 2018. [Online]. Available: https://cert.org.cn/publish/main/44/index.html/

[2] 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.

[3] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "FANCI : Feature-based automated NXDomain classification and intelligence," in *Proc. USENIX Secur. Symp. (USENIX Secur)*, Aug. 2018, pp. 1165–1181.
[4] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through DNS data analysis," *ACM Comput. Surv.*, vol. 51, no. 4, Sep. 2018, Art. no. 67.
[5] H. Gao, V. Yegneswaran, J. Jiang, Y. Chen, P. Porras, and S. Ghosh, "Reexamining DNS from a global recursive resolver perspective," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 43–57, Feb. 2016.
[6] R. Kozik, M. Pawlicki, and M. Choraś, "Cost-sensitive distributed machine learning for netflow-based botnet activity detection," *Secur. Commun. Netw.*, vol. 2018, Dec. 2018, Art. no. 8753870.
[7] M. Mowbray and J. Hagen, "Finding domain-generation algorithms by looking at length distribution," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSRE)*, Nov. 2014, pp. 395–400.
[8] 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, Apr. 2014, Art. no. 14.
[9] R. Perdisci, I. Corona, and G. Giacinto, "Early detection of malicious flux networks via large-scale passive DNS traffic analysis," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 5, pp. 714–726, Sep./Oct. 2012.
[10] K. Alieyan, A. ALmomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," *Neural Comput. Appl.*, vol. 28, no. 7, pp. 1541–1558, 2017.
[11] H. Lin, Y. Li, W. Wang, and Y. Yue, "Efficient segment pattern based method for malixious URL detection," *J. Commun.*, vol. 36, pp. 141–148, Nov. 2015.
[12] K. Lasota and A. Kozakiewicz, "Analysis of the similarities in malicious DNS domain names," in *Proc. Int. Workshop Covergence Secure Perva. Environ. (IWCS)*, Jun. 2011, pp. 1–6.
[13] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *Proc. Int. Workshop Recent Intrusion Adv. Detection*, Oct. 2014, pp. 1–21.
[14] H. Zhao, Z. Chang, and L. Wang, "Fast malicious domain name detection algorithm based on lexical features," *J. Comput. Appl.*, vol. 39, no. 1, pp. 227–231, Mar. 2019.
[15] K. Sato, K. Ishibashi, T. Toyono, H. Hasegawa, and H. Yoshino, "Extending black domain name list by using co-occurrence relation between DNS queries," *IEICE Trans. Commun.*, vol. E95.B, no. 3, pp. 794–802, Mar. 2012.
[16] B. Altay, T. Dokeroglu, and A. Cosar, "Context-sensitive and keyword density-based supervised machine learning techniques for malicious Webpage detection," *Soft Comput.*, vol. 23, no. 12, pp. 4177–4191, Jun. 2019.
[17] D. Huang, K. Xu, and J. Pei, "Malicious URL detection by dynamically mining patterns without pre-defined elements," *World Wide Web*, vol. 17, no. 6, pp. 1375–1394, Nov. 2014.
[18] M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index," *Hum.-Centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 17, Jun. 2017.
[19] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in *Proc. 10th GI Int. Conf. Det. Int. Malware, Vulnerability Assessment (DIMVA)*, Jul. 2014, pp. 192–211.
[20] Q. Hai and S. Hwang, "Detection of malicious URLs based on word vector representation and ngram," *J. Intell. Fuzzy Syst.*, vol. 35, no. 6, pp. 5889–5900, Dec. 2018.
[21] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1663–1677, Oct. 2012.
[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 Secur.*, vol. 19, no. 2, Sep. 2016, Art. no. 4.
[23] L. Bilge, S. Sen, D. Balzarotti, C. Kruegel, and E. Kirda, "Exposure: A passive DNS analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, Apr. 2014, Art. no. 14.
[24] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, A. N. Saeed, and L. Wenke, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *Proc. Usenix Conf. Secur. Symp.*, Aug. 2012, pp. 491–506.
[25] *Alexa Top Global Sites*. Accessed: Jan. 18, 2018. [Online]. Available: https://support.alexa.com/
[26] D. A. Orr and L. Sanchez, "Alexa, did you get that? Determining the evidentiary value of data stored by the Amazon Echo," *Digit. Investig.*, vol. 24, pp. 72–78, Mar. 2018.
[27] L. Carvajal, L. Quesada, G. López, and J. A. Brenes, "Developing a proxy service to bring naturality to Amazon's personal assistant 'Alexa,'" in *Advances in Human Factors and Systems Interaction*. Los Angeles, CA, USA: Springer, 2017, pp. 260–270.
[28] I. Najafi, M. Kamyar, A. Kamyar, and M. Tahmassebpour, "Investigation of the correlation between trust and reputation in B2C e-commerce using Alexa ranking," *IEEE Access*, vol. 5, pp. 12286–12292, 2017.
[29] E. Casalicchio, M. Caselli, and A. Coletta, "Measuring the global domain name system," *IEEE Netw.*, vol. 27, no. 1, pp. 25–31, Jan. 2013.
[30] M. Wang, Z. Zhang, and H. Xu, "DNS configurations and its security analyzing via resource records of the top-level domains," in *Proc. IEEE Int. Conf. Anti-Counterfeiting, Secur, Ident.*, Oct. 2017, pp. 21–25.
[31] W. Quan, C. Q. Xu, J. F. Guan, H. K. Zhang, and L. A. Grieco, "Scalable name lookup with adaptive prefix Bloom filter for named data networking," *IEEE Commun. Lett.*, vol. 18, no. 1, pp. 102–105, Jan. 2014.
[32] Z. Yan, H. Li, S. Zeadally, Y. Zeng, and G. Geng, "Is DNS ready for ubiquitous Internet of Things?" *IEEE Access*, vol. 7, pp. 28835–28846, 2019.
[33] J. Luo and Y. Lepage, "A method of generating translations of unseen n-grams by using proportional analogy," *IEEJ Trans. Electr. Electron. Eng.*, vol. 11, no. 3, pp. 325–330, Feb. 2016.
[34] H. Zhao, Z. Chang, G. Bao, and X. Zeng, "Malicious domain names detection algorithm based on N-Gram," *J. Comput. Netw. Commun.*, vol. 2019, Feb. 2019, Art. no. 4612474.
[35] M. Aman, A. B. M. Said, S. J. A. Kadir, and I. Ullah, "Key concept identification: A sentence parse tree-based technique for candidate feature extraction from unstructured texts," *IEEE Access*, vol. 6, pp. 60403–60413, 2018.
[36] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, A. K. Sangaiah, and F. Martinelli, "Classification of ransomware families with machine learning based on *N*-gram of opcodes," *Future Gener. Comput. Syst.*, vol. 90, pp. 211–221, Jan. 2019.
[37] L. Yang, J. Zhai, W. Liu, X. Ji, G. Liu, Y. Dai, and H. Bai, "Detecting word-based algorithmically generated domains using semantic analysis," *Symmetry*, vol. 11, no. 2, p. 176, Feb. 2019.
[38] Y. Fu, L. Yu, O. Hambolu, I. Ozcelik, B. Husain, and J. Sun, "Stealthy domain generation algorithms," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1430–1443, Jun. 2017.
[39] J. He, P. Flener, and J. Pearson, "Underestimating the cost of a soft constraint is dangerous: Revisiting the edit-distance based soft regular constraint," *J. Heuristics*, vol. 19, no. 5, pp. 729–756, Oct. 2013.
[40] W. Luo and T. Cao, "Malware detection approach based on non-user operating sequence," *J. Comput. Appl.*, vol. 38, no. 1, pp. 56–60, Jan. 2018.
[41] M. Ohsaki, P. Wang, K. Matsuda, S. Katagiri, H. Watanabe, and A. Ralescu, "Confusion-matrix-based kernel logistic regression for imbalanced data classification," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1806–1819, Sep. 2017.
[42] D.-T. Truong, G. Cheng, A. Jakalan, X.-J. Guo, and A.-P. Zhou, "Detecting DGA-based botnet with DNS traffic analysis in monitored network," *J. Internet Technol.*, vol. 17, no. 2, pp. 217–230, Mar. 2016.
[43] *Malware Domain Blocklist*. Accessed: Jan. 18, 2018. [Online]. Available: https://malwaredomains.com/
[44] *Malicious Domain List*. Accessed: Jan. 18, 2018. [Online]. Available: https://malware-domainlist.com/
[45] *ZeuS Tracker: ZeuS Blocklist*. Accessed: Jan. 18, 2018. [Online]. Available: https://zeustracker.abuse.ch/blocklist.php?download=Dom-ainblocklist
[46] R. Weaver, "Visualizing and modeling the scanning behavior of the Conficker botnet in the presence of user and network activity," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 5, pp. 1039–1051, May 2015.
[47] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proc. ACM Conf. Comput. Commun. Secur.*, Nov. 2009, pp. 635–647.
[48] *The DGA of Symmi*. Accessed: Jan. 18, 2018. [Online]. Available: https://johannesbader.ch/2015/01/the-dga-of-symmi/
[49] *Anquan Organization*. Accessed: Jan. 18, 2018. [Online]. Available: https://v.anquan.org/cert/
[50] *PhishTank-Join the Fight Against Phishing*. Accessed: Jan. 18, 2018. [Online]. Available: https://alexa.com/
[51] *The DGA of Newgoz*. Accessed: Jan. 18, 2018. [Online]. Available: https://johann-esbader.ch/2014/12/the-dga-of-newgoz/
[52] *The DGA of Shiotob*. Accessed: Jan. 18, 2018. [Online]. Available: https://johannesbader.ch/2015/01/the-dga-of-shiotob/

HONG ZHAO received the B.S. degree from Northwest Normal University, in 1993, and the Ph.D. degree from Xinjiang University, in 2010. Since 1993, he has been with the School of Computer Science, Lanzhou University of Technology, where he become a Full Professor, in 2010. He has authored four academic books and over 30 refereed articles. His current research interests include deep learning, embedded systems, and natural language processing.

ZHAOBIN CHANG received the B.S. degree from the Lanzhou University of Technology, Lanzhou, China, in 2017. He has authored three refereed articles. His current research interests include cyberspace security, natural language processing, and deep learning.

WEIJIE WANG received the B.S. degree from Harbin Finance University, Harbin, China, in 2016. Her current research interests include deep learning and speaker recognition.

XIANGYAN ZENG received the B.S. degree in computer science and information engineering and the M.S. degree in computer applications from the Hefei University of Technology, China, in 1987 and 1990, respectively, the M.S. degree in electrical and electronics engineering, in 2001, and the Ph.D. degree in computer science from the University of the Ryukyus, Japan, in 2004. She is currently a Professor with the Department of Mathematics and Computer Science, Fort Valley State University. She has authored more than 40 referred articles. Her research interests include computer vision, image processing, pattern recognition, and machine learning.