Neural Process Lett (2018) 48:1347–1357
https://doi.org/10.1007/s11063-017-9666-7

Malicious Domain Name Detection Based on Extreme Machine Learning

Yong Shi[1] · **Gong Chen**[1] · **Juntao Li**[1]

Published online: 3 July 2017

Abstract Malicious domain detection is one of the most effective approaches applied in detecting Advanced Persistent Threat (APT), the most sophisticated and stealthy threat to modern network. Domain name analysis provides security experts with insights to identify the Command and Control (C&C) communications in APT attacks. In this paper, we propose a machine learning based methodology to detect malware domain names by using Extreme Learning Machine (ELM). ELM is a modern neural network with high accuracy and fast learning speed. We apply ELM to classify domain names based on features extracted from multiple resources. Our experiment reveals the introduced detection method is able to perform high detection rate and accuracy (of more than 95%). The fast learning speed of our ELM based approach is also demonstrated by a comparative experiment. Hence, we believe our method using ELM is both effective and efficient to identify malicious domains and therefore enhance the current detection mechanism of APT attacks.

Keywords Advanced Persistent Threat · Domain name · DNS · C&C communication · Extreme Learning Machine

1 Introduction

Recent years have witnessed an exponential growth in the influence of Internet on the daily activities of both organizations and individuals around the globe. High reliance on various web pages and applications not only enhances the proficiency of information transmission

✉ Gong Chen
gchen08@sjtu.edu.cn

Yong Shi
shiyong@sjtu.edu.cn

Juntao Li
nikolaslee@sjtu.edu.cn

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, 800, Dongchuan Rd., Minhang District, Shanghai 200240, People's Republic of China

but also increases the vulnerability to cyber criminals. Cybercrime is a relatively low-risk crime but the detrimental impact caused by malicious behaviors is severe.

On one hand, technological innovation provides security experts with powerful and multi-functional intrusion detection systems (IDSs) against cyber threats for decades. On the other hand, the advanced network techniques are also taken advantages by hackers to execute stealthy and effective attacks.

Advanced Persistent Threats (APTs), also known as targeted attacks, are considered to be the most complicated and atrocious cyber threats. The victim of an APT is a preselected organization or enterprise, which will suffer from long-term penetrative and stealthy attacks over time. Multiple attack techniques, such as phishing, social engineering, malware and also backdoor program, are practiced within the life cycle of a successful APT attack [1]. The diversity of attack techniques, the complex of malware tools and the sophistication of a well-organized campaign behind an APT make the threat much hard to be detected or traced. However, if security specialists can identify malware Command and Control (C&C) communications, which plays an integral role as the bridge between attacks and compromised devices, malicious operations can never remain covert and undetectable. C&C communication requires Domain Name System (DNS) as its backbone to implement malware infection and data exfiltration in most cases [2]. Therefore, DNS analysis was proposed to be a significant and promising detection technique on BlackHat 2014 USA [3]. This paper proposes a method for determining malicious domains which could be used as a supplement in detecting APT attacks.

The rest of this paper is structured as follows. Section 2 discusses previous literature related to domain name detection. Section 3 describes the methodology that we follow to conduct the analysis. Section 4 presents the evaluation of our approach's performance and the conclusion is given in Sect. 5.

2 Related Work

2.1 Malicious Domain Generation

Botnets, comprised of a group of vulnerable computers called "bot", are widely used by hackers to perform C&C communications. To avoid detection and obfuscate tracking, intruders apply two DNS techniques, Fast-Flux and Domain-Flux, to hide their true C&C servers [4]. Fast-Flux aims to associate a fully qualified domain with IP addresses in great quantities. By assigning a relatively low time-to-live (TTL) to each IP and swapping IPs in and out of flux frequently, attackers are able to change the DNS logs and ultimately associate the absolute domain name with a set of different IPs. The basic idea of Domain-Flux technique is to hide the malicious domain names of its C&C servers behind plentiful domains generated by Domain name Generation Algorithm (DGA). DGA can generate a combination of alphanumeric letters as a random domain based on a seed (which usually is the current date and time) [5]. These evasive techniques render the traditional security policies ineffective, including domain blacklisting, restriction on IP ranges or other signature based approaches.

2.2 Domain Name Detection

Passive DNS analysis is proved to be a typical and practicable detection method because it provides security researcher with data to characterize domain names and investigate their behaviors. According to previous work, there are three common approaches: blacklist based,

graph based and machine learning based. Blacklist based approaches rely solely on a list of malicious domains updated by security experts periodically. Blacklist extension method is proposed by Sato et al. based on the co-occurrence relationship between sets of domains [6]. However, these approaches share the limitation that a blacklist can never cover all malicious domains and the amount of time spent in maintenance is always large. Graph structure has been recently applied into modern detection [7–9]. Manadhata et al. propose to frame a host-domain graph from an enterprise's event logs and use graph-based analysis to solve the detection problem [10]. Lee et al. introduce GMAD, graph-based malware activity detection, in which DNS query sequences are modeled as a domain-name-travel graph to detect malicious DNS activities [11]. Chau et al. propose to construct an undirected and unweighed bipartite graph with millions of nodes representing machines and files [12]. Machine learning is a prevalent technique applied in industries to deal with classification, clustering or data mining [13–15]. Malicious domain detection can be seen as a kind of pattern recognition problem where machine learning can serve as an effective method. After trained by sets of existing domain names, machine learning systems can be leveraged to identify new incoming domains based on multiple selected features. Zou et al. use J48 decision tree as the classification algorithm and implement the combination of real-time detection and long-term monitoring [16]. Decision tree is also applied in EXPOSURE, a detection system presented by Bilge et al [17]. Amini et al. propose a clustering scheme to find group similar traffic patterns based on flow features in NetFlow protocol records [18]. Yu et al. use the weighted support vector machine (SVM) to discriminate benign from malicious domains [19].

Previous researches on malware domain detection usually faced the problem that the processing speed of analyzing large-scale data limits the scalability and efficiency. In this paper, our detection mechanism is to characterize a given domain using several extracted features and the classification algorithm is based on Extreme Learning Machine (ELM). Compared with other methods, ELM has the advantage of good generalization ability and fast learning speed, which greatly facilitate the detection progress [26].

3 Methodology

This section gives a description of the methodology adopted in our work, including the details of the features we choose to differentiate domain names and the mechanism of our classifier based on ELM.

3.1 Feature Selection

The features that we select to characterize a domain name are classified into four categories: construction-based, IP-based, TTL-based and WHOIS-based features. The features are summarized in Table 1 and the details are described in the rest of this section.

Construction-based features are characteristics extracted from the domain name itself to describe the structural and lexical properties. DNS is equivalent to a phone book in the Internet and each domain name serves as the "nickname" of an unique IP address. The benign domain names are usually readable because web owners hope their domains are easy for people to memorize and type. However, attackers are never concerned with how well their domains are constructed but how many candidate domains they can possess. Malicious domain names share some similarities in construction especially for the domains generated by DGA [20]. Therefore, we select the following three features as construction-based features and we believe they are indicative to identify malicious domains.

Table 1 Features indicative of malicious behaviors

No.	Feature name	Classification
1	Length of domain	Construction-based features
2	Number of consecutive characters	
3	Entropy of domain	
4	Number of IP addresses	IP-based features
5	Number of countries	
6	Average TTL value	TTL-based features
7	Standard deviation of TTL	
8	Life time of domain	WHOIS-based features
9	Active time of domain	

Feature 1: Length of Domain The average length of malicious domains is longer than that of non-malicious domains according to previous studies [21]. Legitimate web owners are aware of the significance to pick a succinct domain because every additional letter would increase the chance for users to misremember the correct domain name. In contrast, the amount of malicious domains randomly generated by DGA is always large and therefore the average domain name length is longer than benign domains. Hence, we formulate the first feature of a given domain $Domain^{(i)}$ as

$$Feature_1 = length\{Domain^{(i)}\} \tag{1}$$

Feature 2: Number of Consecutive Characters Alphabetical sequences of more than three repeated letters are not common in English words as well as benign domain names. However, domains produced by DGA tend to contain more consecutive repeated characters. Note that we only focus on the consecutive sequences of letters and hyphens because repeated numerical characters are sometimes seen in benign domains (e.g., www.10000.com). We take the maximum number of reduplicate characters contained in $Domain^{(i)}$ as its second feature.

$$Feature_2 = max\{\#\ of\ consecutive\ repeated\ characters\ in\ Domain^{(i)}\} \tag{2}$$

Feature 3: Entropy of Domain A malicious domain based on Domain-Flux is a random combination of numerical, alphabetical characters and hyphens. The randomness in construction can be leveraged as an indication to differentiate malicious domains from legitimate ones. Illuminated by the entropy in information theory, we introduce the entropy of a domain as a measure of the disorder or chaos of a domain's structure [22]. We calculate the entropy (i.e., Feature 3) of a give domain $Domain^{(i)}$ consisting of n_i distinct characters $\{c_1^i, c_2^i, \cdots, c_{n_i}^i\}$ by

$$Feature_3 = -\sum_{j=1}^{n_i} p_j^i \times lb(p_j^i) \tag{3}$$

where $p_j^i = count(c_j^i)/length(Domain^{(i)})$ is the probability of character c_j^i, lb is the logarithm of base two.

DNS provides a one-to-many mapping between domain names and IP addresses, which is known as DNS records of type "A". In a botnet especially a Fast-Flux Service Network

(FFSN), whenever a new compromised "bot" is added or a fault one is deleted, the list of IP addresses contained in DNS "A" record is updated [23]. The pattern of DNS "A" records in a botnet can be identified by two characteristics. The first is that a malicious domain is typically resolved to multiple IP addresses because miscreants use Fast-Flux as an evasive technique to hide from tracing. The second is the diversity of IP addresses resolved from a specific domain name. The following two features we use to distinguish between malware and legitimate domains are based on the corresponding IP addresses.

Feature 4: Number of IP addresses A domain can be associated with multiple IP addresses especially for big corporations. In the cases that some companies or organizations adopt round robin DNS for load distribution, the domain-IP mapping will change periodically [24]. However, through empirical observation we find the number of IP addresses mapped to malicious domains is larger than non-malicious domains. Therefore, the number of distinct IP addresses is identified as as Feature 4.

$$Feature_4 = count\{distinct\ IP\ addresses\} \tag{4}$$

Feature 5: Number of countries The IPs belonging to a Autonomous System Number(ASN) or a organization usually have similarities as their locations are restricted in certain places. In a botnet, compromised "bot" are most likely to be geographically diverse in order to help the centralized C&C servers elude detecting and tracking. The diversity can be observed by carefully investing the locations of IP addresses resolved from a malicious domain. Hence, We extract the number of distinct countries (or regions) of IPs from IP lookup information as Feature 5.

$$Feature_5 = count\{distinct\ countries\} \tag{5}$$

Time-to-live (TTL) in a DNS record determines how long the resource record of a corresponding domain should be cached before updated. Associating low TTL value to malicious domains is usually leveraged by attackers to abuse DNS blacklisting. By setting low TTL value, attackers can force the non-authoritative name servers to flush caches and query to the authoritative server at high frequencies, which enhances the update procedure of botnet. We extract the following two features based on TTL values in DNS records.

Feature 6: Average TTL value Commonly, TTL value for DNS was set to be 86,400 s (i.e., 24 h) for high resolution speed, but in recent years more DNSs prefer lower TTL value for availability. Though it can not be conclude that the lower TTL value of DNS records is, the higher the possibility of the corresponding domain to be malware is, the average TTL value associated with legitimate domains tends to be significantly higher than that with malicious domains [17]. The expression of Feature 6 is as follows.

$$Feature_6 = average\{TTL\ in\ DNS\ records\ of\ Domain^{(i)}\} \tag{6}$$

Feature 7: Standard Deviation of TTL The TTL values set for malicious domains are observed to be limited in a relatively small range (e.g., mostly less than 100 s in [17]). In contrast, the distribution of TTL values from records for benign domains covers a large span, because various authoritative name servers set their own TTL values for different purposes. Hence, we propose to use the standard deviation of TTL as Feature 7.

$$Feature_7 = standard\_dev\{TTL\ in\ DNS\ records\ of\ Domain^{(i)}\} \tag{7}$$

ICANN's WHOIS lookup service provides a public access to information called "WHOIS data" beyond domain names, including the date of registration, expiration, update and the details of registrant [25]. WHOIS properties have been used to analysis domain names in previous work [21,23]. In this study, we only fucus on time-based information extracted from WHOIS queries based on the facts that attackers are most likely to use fake identities to complete the registration. Two features extracted from WHOIS data are presented as follows.

Feature 8: Life Time of Domain Benign web owners usually register their domains for long-term business while malicious domains are typically of much shorter life time (i.e., age). Domains utilized by miscreants would be deactivated once detected to be malicious. We take the interval (counted in days) between Registration Expiration Date and Created Date from WHOIS data as Feature 8.

$$Feature_8 = Date_{Expiration} - Date_{Created} \tag{8}$$

Feature 9: Active Time of Domain With the insights presented above, the active time of domains for malicious purpose is considered to be short. Whenever old domains are deactivated by authorities, attackers would register new ones rapidly and employ them for malicious purpose before detected and blocked by authorities. Similar to the life time of domain, we propose the active time of domain as Feature 9.

$$Feature_9 = Date_{Updated} - Date_{Created} \tag{9}$$

3.2 Extreme Learning Machine

A lot of machine learning techniques are applied to analyze and classify domain names. However, the usage of neural network is rarely seen in previous researches because the slow learning speed limits the performance in detection problems. Extreme Learning Machine(ELM) proposed by Huang et al. is a new learning scheme for Single-hidden-Layer Feedforward neural Networks (SLFNs) with fast learning speed [26,27].

We module the detection problem as a SLFN, where the neuron (e.g., nodes or samples) is the representative of a domain to be identified. Let n be the number of neurons in a training set with k distinct classes, where each neuron is characterized by m different features. Hence, the training set is denoted as

$$Training\_Set = \{(x^{(i)}, t^{(i)}) | 1 \leq i \leq n\} \tag{10}$$

where $X^{(i)} = (x^{(i)}, t^{(i)})$ is the ith training sample, $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}]^T$ is the feature vector and $t^{(i)} = [t_1^{(i)}, t_2^{(i)}, \ldots, t_k^{(i)}]^T$ is the output vector of classification. According to [26], a SLFN with L hidden neurons is mathematically denoted as

$$\sum_{j=1}^{L} \beta_j g(W_j \cdot X^{(i)} + b_j) = o_i \tag{11}$$

where β_j is the output weight vector of jth hidden node, $g(x)$ is the activation function, W_j is the input weight vector, b_j is the threshold and o_i is the output value of i^{th} neuron. The objective of ELM is to minimize $\|H\beta - T\|$, where

$$H = \begin{bmatrix} g(W_1 \cdot X^{(i)} + b_1) & \cdots & g(W_L \cdot X^{(1)} + b_L) \\ \vdots & \cdots & \vdots \\ g(W_1 \cdot X^{(n)} + b_1) & \cdots & g(W_L \cdot X^{(n)} + b_L) \end{bmatrix} \tag{12}$$

represents the effect of hidden layer, $\beta = [\beta_1^T, \cdots, \beta_L^T]^T$ is the matrix of output weights and $T = [T_1^T, \ldots, T_N^T]^T$ is the estimated output. The objective is equivalent to minimizing the cost function

$$E = \sum_{i=1}^{n} \left(\sum_{j=1}^{L} \beta_j g(W_j \cdot X^{(i)} + b_j) - t^{(i)} \right)^2 \tag{13}$$

which can be solved by repeatedly adapting variables in each irritation until convergence in traditional machine learning techniques like gradient descent or Back Propagation (BP) algorithms. However, ELM regards $\|H\beta - T\| = 0$ as a linear system by randomly choosing W_j and b_j for hidden nodes. The linear system can be solved by simply evaluating output weight as $\widehat{\beta} = H^{\dagger}T$ where $H^{\dagger}$ is the $Moore - Penrose$ generalized inverse of H. Hence, ELM can perform the learning process in a relatively fast speed.

4 Experiments

4.1 Data Generation

The data used in this paper was collected from the DNS queries received by five DNS servers in Network and Information Center of Shanghai Jiaotong University, which deals with 3000 queries per second on average. We obtained a total of 9,335,270 queries by monitoring the traffic for a week.

To reduce the scale of traffic data, we carried out a filtering process to generate a more reliable and manageable data set of benign and malware domains. First, the domains of which the Alexa ranking is within top 100,000 (for recent 3 months) were identified as legitimate [28]. These popular domains are always well-maintained and are not likely to be leveraged by miscreants. Second, we blacklisted the domain names reported to be malicious by Malicious Domain List and Phishing Tank. [29,30]. Third, we eliminated those invalid domains without any WHOIS information. Attackers typically registered many domains generated by DGA to hinder authorities from detecting their C&C communications. Therefore, a closed domain is neither valuable to attackers nor to our classifier's construction. Note that we did not filter out the domains with incomplete WHOIS data (i.e. their WHOIS properties contain at least one but not all of Created Date, Registration Expired Date and Updated Date). Actually, many registrars do not fill or update complete WHOIS information in real-world, so keeping these domains in our data set is of practical value. After the filtering process, the volume of data was reduced and the final data set consists of 12,096 benign domains and 38,915 malware domains.

4.2 Results and Discussion

To evaluate the performance of our detection method, we set malicious domains as positive instances and benign domains as negative. We adopted detection rate and accuracy as the main metrics, defined as

$$DetectionRate = TP/P \tag{14}$$

$$Accuray = (TP + TN)/(P + N) \tag{15}$$

Table 2 Detection rate and accuracy of ELM

No. of nodes	Training time (s)	Testing time (s)	Detection rate (%)	Accuracy (%)
10	0.03	0.01	92.42	95.01
20	0.07	0.03	95.28	95.34
50	0.24	0.08	95.90	95.75
100	0.72	0.15	94.67	94.67
200	1.75	0.28	94.27	96.27
300	3.07	0.42	94.32	96.28
500	6.24	0.60	94.27	96.29
1000	19.62	1.12	93.31	96.16
1500	41.09	1.59	92.80	96.02

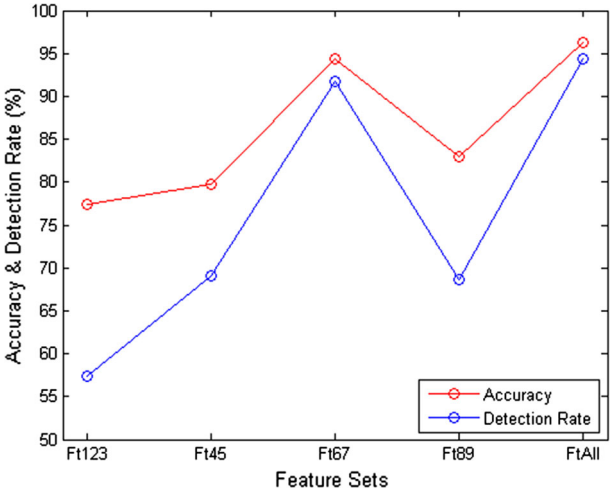

Fig. 1 Accuracy and detection rate of different feature sets

where T and P are the number of positive and negative instances respectively, TP is True Positive (i.e. the number of malicious domains identified as malware) and TN is True Negative (i.e. the number of benign domains identified to be legitimate).

We performed the experiments using Matlab R2011b on a Windows 10 64 Bit PC with 2.40 GHz Intel Quad Core and 8G of RAM. In order to mitigate the instability of ELM (due to the randomness of weight between hidden neurons and input layer), we applied 10-fold cross-validation and 50% percent split (i.e. half domains of our data set were randomly selected to constitute the training set and the rest were used for testing the classifier). The average of outcomes in all runs was received as the result. We opted the number of hidden nodes and results of the experiment is summarized in Table 2.

As shown in Table 2, we achieved the detection rate of 95.90% and the accuracy of 96.29% when the number of hidden nodes is set to be 50 and 500 respectively. Note that the training and testing time grow with the number of hidden nodes, while the detection rate and accuracy tend to remain high stably. This implies that we can reduce the number of neurons for faster learning speed but still obtain high accuracy and precision. Based on this observation, we set the number of hidden nodes to be 300 as default for further experiments.

Table 3 Comparison of different classifier

Classifier	Training time (s)	Testing time (s)	Accuracy (%)
LR	0.02	0.03	91.95
CART	0.02	0.01	91.83
BPNN	41.95	0.31	95.82
SVM	88.17	0.90	94.70
ELM	3.07	0.42	96.28

As mentioned previously, the 9 features we propose in Table 1 are classified into 4 distinct classes with details described in Chapter 3.1. To validate the effectiveness of the features, we compared the results of ELM running on different feature sets. We use "Ft123", "Ft45", "Ft67" and "Ft89" to denote the features based on construction, IP, TTL and WHOIS information respectively; "FtAll" denotes the combination of all feature sets. Figure 1 illustrates the features based on TTL value produce higher accuracy and detection rate than other classes. By manually investing the misclassified domains from our data set, we found that part of malicious domains are idiomatically designed. It was also found that some of benign domains lack complete WHOIS information. These facts result in that the lexical features and WHOIS-based features contribute less to the classification procedure. In contrast, DNS records of most malicious domains generated by DGA share a common attribute: relatively low TTL value, which makes the TTL-based features become a more distinguished characteristic related to malicious activities. However, ELM generates the best performance when all the features are combined. The features we extracted are certainly not a complete description of a domain, but the result demonstrates the proposed features are instructive and effective to detect malicious domains.

We further used four different classifiers: LR (Logistic Regression), CART (Classification and Regression Tree), BPNN (Back Propagation Neural Network) and SVM (support vector machine). Note the efficiency and precision of a classifier depend greatly on its parameters. For example, the more the number of hidden neurons in a BPNN, the better the fitting precision but the longer the training time. Therefore, we applied trail and error to choose the proper value of parameters for every classifier in our comparative experiment. For BNN, we constructed a single-hidden-layer BP neural network with 10 hidden neurons and limited the value of maximum iteration to 500. For SVM, we averaged the results of 10 trails as the parameters of SVM are randomly chosen. Our experimental outcome is summarized in Table 3. Although the computational efficiency of LR and CART is excellent, the accuracy is not of acceptable standard. BPNN and SVM can achieve high accuracy but the progress of training is comparably time consuming due to their computational complexity. Compared with other classification techniques, ELM has the superior performance in both accuracy and learning speed.

5 Conclusions

Domain name analysis is proposed to be an effective and significant approach to detect APTs. However, in order to avoid detection and elude tracking, evasive techniques like DGA are widely leveraged by modern perpetrators to generate a large number of malware domains, which poses serious challenges faced with current detection systems. High accuracy

or precision is no more the single primary goal for a detection system and how to improve the efficiency has also become a critical issue.

In this paper we present a detection method towards the detection for malicious domains. Nine features classified into four categories (Construction-based, IP-based, TTL-based and WHOIS-based) are identified to characterize domain names and ELM is utilized as the classifier in our methodology. The outcome of our experiment demonstrates that the features we propose are indicative to associate domain names with malicious activities. According to further analysis of our comparative experiment, ELM not only yields good performance in detecting malicious but shows a clear advantage of fast learning speed. We believe our research can reveal some representative patterns of malware domains and be exploited as an effective supplement to the existing approaches for detecting APT attacks.

References

1. Ghafir I, Prenosil V (2014) Advanced persistent threat attack detection: an overview. Int J Adv Comput Netw Secur 4:50–54
2. Li M, Huang W, Wang Y, Fan W, Li J (2016) The study of APT attack stage model. In: 2016 IEEE/ACIS 15th international conference on computer and information science (ICIS), pp 1–5
3. Li F APT attribution and DNS profiling. http://www.blackhat.com/docs/us-14/materials/us-14-Li-APT-Attribution-And-DNS-Profiling-WP.pdf
4. Soltani S, Seno SAH, Nezhadkamali M, Budiarto R (2014) A survey on real world botnets and detection mechanisms. Int J Inf Netw Secur 3:116–127
5. Grill M, Nikolaev I, Valeros V, Rehak M (2015) Detecting DGA malware using NetFlow. In: 2015 IFIP/IEEE international symposium on integrated network management (IM). IEEE, pp 1304–1309
6. Sato K, Ishibashi K, Toyono T, Miyake N (2012) Extending black domain name list by using co-occurrence relation between DNS queries. IEICE Trans Commun 95:794–802
7. Zhang S (2014) Detecting malware domains on DNS traffic. Master Thesis, Shanghai Jiaotong University
8. Shi L, Lin D, Fang CV, Zhai Y (2015) A hybrid learning from multi-behavior for malicious domain detection on enterprise network. In: 2015 IEEE international conference on data mining workshop (ICDMW). pp 987–996
9. Gao Y, Zhen Y, Li H, Chua TS (2016) Filtering of brand-related microblogs using social-smooth multiview embedding. IEEE Trans Multimed 18:2115–2126
10. Manadhata PK, Yadav S, Rao P, Horne W (2014) Detecting malicious domains via graph inference. In: European symposium on research in computer security. Springer, pp 1–18
11. Lee J, Lee H (2014) GMAD: graph-based malware activity detection by DNS traffic analysis. Comput Commun 49:33–47
12. Chau DH, Nachenberg C, Wilhelm J, Wright A, Faloutsos C (2010) Polonium: Tera-scale graph mining for malware detection. In: Acm sigkdd conference on knowledge discovery and data mining
13. Gao Y, Zhang H, Zhao X, Yan S (2017) Event classification in microblog via social tracking. ACM Trans Intell Syst Technol 8:1–14
14. Ding G, Guo Y, Zhou J, Gao Y (2016) Large-scale cross-modality search via collective matrix factorization hashing. IEEE Trans Image Process 25:5427–5440
15. Mashechkin IV, Petrovskii MI, Tsarev DV (2016) Machine learning methods for analyzing user behavior when accessing text data in information security problems. Mosc Univ Comput Math Cybern 40:179–184
16. Futai Z, Siyu Z, Weixiong R (2013) Hybrid detection and tracking of fast-flux botnet on domain name system traffic. China Commun 10:81–94
17. Bilge L, Kirda E, Kruegel C, Balduzzi M (2011) EXPOSURE: finding malicious domains using passive DNS analysis. In: Network and distributed system security symposium
18. Amini P, Azmi R, Araghizadeh M (2014) Botnet detection using NetFlow and clustering. Adv Comput Sci Int J 3:139–149
19. Yu X, Zhang B, Kang L, Chen J (2012) Fast-flux botnet detection based on weighted svm. Inf Technol J 11:1048–1055
20. Lasota K, Kozakiewicz A (2011) Analysis of the similarities in malicious DNS domain names. In: International conference on secure and trust computing, data management, and application, 1006

21. Ma J, Saul LK, Savage S, Voelker GM (2009) Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1245–1254
22. Shannon CE (2001) A mathematical theory of communication. ACM SIGMOBILE Mob Comput Commun Rev 5:3–55
23. Passerini E, Paleari R, Martignoni L, Bruschi D (2008) Fluxor: detecting and monitoring fast-flux service networks. In: International conference on detection of intrusions and malware, and vulnerability assessment. pp 186–206
24. Brisco T DNS support for load balancing. https://tools.ietf.org/html/rfc1794
25. ICANN WHOIS: WHOIS Search. https://whois.icann.org/en
26. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
27. Huang GB (2015) What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. Cogn Comput 7:263–278
28. Website Traffic, Statistics and Analytics—Alexa. http://www.alexa.com/siteinfo
29. Malicious Domain List. https://www.malwaredomainlist.com/
30. PhishTank—Join the fight against phishing. http://www.alexa.com/