Fast-Flux Malicious Domain Name Detection Method Based on Domain Resolution Spatial Features

Shaojie Chen[1][a], Bo Lang[1,2] and Chong Xie[1]

[1] *State Key Laboratory of Software Development Environment, Beihang University, Beijing, China*

[2] *Zhongguancun Laboratory, Beijing, China*

Keywords: Fast-Flux Domain Name Detection, Domain Resolution Spatial Features, Resolution Spatial Relationship Graph, GCN, Botnet.

Abstract: Fast-Flux malicious domain names evade detection by quickly changing the resolved IP addresses of the domain name, and play an important role in cyberattacks. In order to improve the performance of the Fast-Flux domain name detection, this paper explores and uses the rich spatial features contained in the domain name resolution process, and proposes a Fast-Flux malicious domain name detection method based on the domain resolution spatial features. In this method, the CNAMEs and IPs in the resolution results obtained by multiple requests are used as nodes to construct the resolution spatial relationship graph (RSRG). Then the NS record of the second-level domain name, Geographical locations and Autonomous System Numbers of the resolved IPs, and WHOIS information of the domain name are further extracted as the node features in the RSRG. Finally, a GCN model with Max Pooling algorithm is used to extract spatial features from RSRG and perform classification. Our method achieves an accuracy of 94.98% and an F1 value of 92.02% on the self-constructed dataset, and the overall performance is significantly better than the current best methods.

1 INTRODUCTION

Botnet infects a large number of hosts and uses malwares to form a one-to-many control network between the Command and Control (C&C) server and the infected hosts. Early botnets usually directly wrote the domain name or Internet Protocol (IP) address (abbreviated as IP) of the C&C server in the malwares to establish communications between the infected hosts and the C&C server, which make it easy to be discovered. In order to improve the concealment of the C&C server, attackers use the Domain Name System (DNS) protocol to build Fast-Flux service Networks (FFSNs). In an FFSN, the C&C server controls multiple infected hosts as relay servers, and quickly updates the IPs of the malicious domain name that belongs to the C&C server. In this way, the malicious domain name is resolved to different relay servers, thereby hiding the real C&C server. The malicious domain names used in the FFSNs are called the Fast-Flux malicious domain names. The hidden C&C servers could perform various types of attacks, such as DDoS, spam, and espionage, causing huge losses to organizations and individuals. Therefore, it is of great significance to detect the Fast-Flux domain names for discovering cyberattacks.

The traditional detection method of malicious domain names is to use a blacklist or whitelist; however, FFSN rapidly changes the IPs, making the blacklist ineffective. Compared with domain names generated by Domain Generation Algorithm (DGA) which show obvious text randomness, Fast-Flux domain names show relatively weak randomness in the domain name text, making it difficult to be detected only based on the domain name itself. The DNS protocol data is important for the detection of Fast-Flux domain names. According to the acquisition of data used for detection, detection methods are mainly divided into passive detection methods and active detection methods.

The passive detection methods are generally based on real-time or historical DNS data obtained from the actual network. Such methods do not generate additional traffic in the network, which is not easy to be discovered by attackers. However, such methods usually need to analyze a large amount of historical data, resulting in a large amount of computation, and the attack cannot be detected before the FFSN infects the monitored devices (Zhauniarovich et al., 2018). The

[a] https://orcid.org/0000-0002-1974-6019

Chen, S., Lang, B. and Xie, C.
Fast-Flux Malicious Domain Name Detection Method Based on Domain Resolution Spatial Features.
DOI: 10.5220/0011872700003405
In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP 2023)*, pages 240-251
ISBN: 978-989-758-624-8; ISSN: 2184-4356

active detection methods generally obtain DNS data through actively resolving the domain name to perform detections, and some methods additionally acquire information from third-party databases. Therefore, active detection methods do not rely on large amounts of passive DNS traffic, and could complete the detection process with a single domain name. Although it would cost a certain amount of time to actively acquire the DNS and third-party data, active methods can obtain data for the domain names in a targeted manner, thus effectively reduce the amount of data processing. In addition, such methods could detect the domain names before their actual malicious uses; and do not use network traffic of individual users, which effectively protect user's privacy (Zhauniarovich et al., 2018). Therefore, active detection methods have attracted much attention of researchers, and this paper also researches effective active detection method.

For current active detection methods, the acquisition sources of features are divided into the following three types: 1) Actively obtain resolved IP information of domain names, such as resolving through the *dig* command; 2) Acquire information from third-party databases, such as IP locations, WHOIS, and results from search engine, etc.; 3) Actively access the resolved IPs of the domain names to obtain information, such as response time of the IPs, data transmission paths obtained by *traceroute* command, etc. Among them, the information from the third type, which are obtained by actively access the resolved IPs, will be affected by the network fluctuations; and such methods will leave a large amount of data on the corresponding IPs, which make it easy to be discovered by attackers. Therefore, this paper mainly conducts detection based on the former two types of information mentioned above. In addition, compared with traditional machine learning models, deep learning models could discover potential features. Therefore, we study deep learning methods for Fast-Flux domain name detection.

Fast-Flux malicious domain name is usually resolved to a large number of changing IPs, and these IPs are widely distributed, which usually shows differences from normal domain names in the spatial relationships. Therefore, we propose to perform detection based on the resolution spatial features of domain names. The resolution of a domain name is a complex process, and the information returned from the resolution includes Canonical Name (CNAME), resolved IPs, etc. Such information could be presented as a radial structure centered on the domain name, which contains rich spatial information. Therefore, we first request to resolve the IPs of the domain name, and build a relationship graph based on the returned CNAMEs, IPs, and their dependencies in resolution results. Here, we name the relationship graph as the resolution spatial relationship graph (RSRG). In order to measure the changes of resolved IPs, and to distinguish Fast-Flux domain names from Content Delivery Network (CDN) domain names more effectively, referring to the idea of Chen et al. (Chen et al., 2019), we obtain the resolution information from two different cities for five times respectively, to construct the final RSRG. In addition, Fast-Flux domain names are usually different from normal domain names in the distribution of authoritative nameservers (NS), geographical locations (Geo) and autonomous system numbers (ASN) of resolved IPs, and WHOIS information of domains, etc. We additionally obtain such information to construct 16-dimensional features for each node in the RSRG. Finally, the graph convolutional networks (GCN) with Max Pooling algorithm is used to extract the spatial features from the RSRG, and perform classification to obtain the final detection results.

To summarize, our method has the following contributions:

1) This paper proposes for the first time to detect Fast-Flux malicious domain names based on the domain resolution spatial features. The topology of resolution spatial relationship graph (RSRG) is first constructed based on the CNAME and IP records in the resolution results. Then nameserver and WHOIS information of the domain name, and Geographical locations and Autonomous System Numbers of the resolved IPs are obtained. They are then be used as the node features of the RSRG, to describe the spatial information of the nodes from multiple perspectives. Finally, the graph convolutional networks (GCN) is used to extract domain resolution spatial features from RSRG for classification.

2) We construct a dataset, which requests the resolved IPs from different locations for multiple times, to differentiate Fast-Flux and CDN domain names and measure the changes of the resolved IPs. Our method achieves an accuracy rate of 94.98% and an F1 value of 92.02% on the dataset, which is significantly higher than the existing active detection methods.

The remainder of the paper is organized as follows. Section 2 introduces the related work of Fast-Flux domain name detection, and mainly focuses on active detection methods. Section 3 introduces the motivation and Fast-Flux domain name detection framework based on domain resolution spatial features. Section 4 shows the construction of the datasets and the experiments of the model on the datasets, and compares our method with the existing methods. Section 5 summarizes the paper.

2 RELATED WORK

In recent years, researchers have conducted lots of methods on Fast-Flux malicious domain name detection. These methods can be mainly divided into passive and active detection methods, according to the acquisition manner of the detection data. Passive detection methods are designed to detect Fast-Flux traffic from actual passive traffic, and usually extract features from a single packet or packet set for detection. Active detection methods aim to determine whether a given domain name is a Fast-Flux domain name. These methods usually need to actively obtain the resolution result of the domain name, or additionally obtain data from third-party databases, such as WHOIS and IP geographical locations, and extract features from them for detection. The two types of detection methods are introduced respectively in the following.

(1) Active Detection Methods.

Active detection methods usually actively acquire information from third-party databases. Common data sources include actively obtained DNS data, additional information provided by third-party databases, and information obtained by actively accessing the resolved IPs of the domain names.

Some methods perform detection only based on actively obtained DNS data (Chen et al., 2019; Holz et al., 2008; Berger et al., 2016). The FCDR methods proposed by Chen et al. (Chen et al., 2019) obtained domain name resolution results from two different DNS servers for 5 times respectively, and retained 1 CNAME record and 2 IP records for each resolution result. For each time of request, 2 CNAME and 4 IP records from two DNS servers were combined to form a 102-dimensional feature vector. Finally, a feature sequence with length of 5 was constructed, and an LSTM model was used for classification. This method highlighted the differences between Fast-Flux and CDN domain names through requests from two different DNS servers, and effectively measured the changes of IPs through 5 requests. Berger et al. (Berger et al., 2016) used DNSMap to extract the corresponding relationships between FQDNs and IPs that have changed within a certain period of time, constructed a bipartite graph between FQDNs and IPs, and obtained the agile groups of FQDNs and IPs with pruning method to perform detection.

In addition to DNS data, there are methods obtain additional information, such as geographical locations, ASNs, WHOIS, or results returned from search engines as detection basis. The FluxOR method proposed by Passerini et al. (Passerini et al., 2008) extracted 9-dimensional features of each domain name, including domain name registration features, network availability features, and agent heterogeneity features, and then used a Naive Bayes Classifier to perform classification. Huang et al. (Huang et al., 2010) detected Fast-Flux domain names based on the spatial relationships of resolved IPs. They obtained the location information of each resolved IP obtained from the DNS packets, and calculated 6-dimensional features, such as time zone entropy and the average of minimal service distances, and then classified them based on Bayesian network. Al-Duwairi et al. (Al-Duwairi et al., 2015) used the Google search engine to detect Fast-Flux domain names. They first obtained the resolved IPs of the domain name, searched each IP through the Google search engine, and regarded the numbers of hits as features for detection. In the PASSVM method proposed by Al-Duwairi et al. (Al-Duwairi et al., 2021), the *Censys* (Durumeric et al., 2015) was used to obtain the opening port numbers of the IPs, etc., which are named censys-based features, and obtained the IP geolocation-based and DNS response-based features. They then combined the third parts of features, and perform classification with Support Vector Machine (SVM).

In addition, some studies try to access the resolved IPs of the domain names to obtain features, such as response time (Cafuta et al., 2018; Hsu et al., 2014; Lin et al., 2013; Chandavarkar et al., 2018; Nagunwa et al., 2022). Cafuta et al. (Cafuta et al., 2018) used the network delay and server processing time as the basic features, and additionally obtained results from search engine to detect Fast-Flux domain names. Hsu et al. (Hsu et al., 2014) believed that the response time of each visit to the Fast-Flux domain name would be fluctuating, they made multiple requests to the resolved IPs to obtain the response time, and then calculated a value named *FF-Score* and performed classification with a threshold. Nagunwa et al. (Nagunwa et al., 2022) integrated various types of features, they extracted 6 types of features including temporal features, spatial features, and DNS features, etc., and perform classifications with various machine learning models.

Overall, current active detection methods are usually based on actively obtained DNS data and third-party databases. The features obtained by actively accessing the resolved IPs of the domain name may be interfered by network fluctuations, and such methods will leave access records on the relay servers of the FFSN, making it easier to be discovered by attackers. In addition, most of the current detection methods extract features based on the feature engineering, and perform detection based on rules or traditional machine learning methods, while seldom use deep learning models. However, deep learning models could discover potential features. Therefore, our method is based on the actively obtained DNS data and information from third-party databases, trying to use deep learning methods to extract the potential spatial features of domain names to improve the detection effect.

(2) Passive Detection Methods.
Passive detection methods generally use the actual DNS traffic as the detection content. These methods usually extract the internal features of a single packet or packet set for detection; some passive methods used additional databases to extract features as a supplement to the traffic features for detection.

For methods only use the features of traffic itself (Rana and Aksoy, 2021; Almomani, 2018; Truong et al., 2020; Wang and Chen, 2019a), to ensure the real-time detection performance, some methods only use a single packet for detection. Rana et al. (Rana and Aksoy, 2021) used the TCP/IP packet of DNS protocol as the detection unit. They extracted features from TCP/IP packet headers, and obtained best feature selection with genetic algorithm. Finally, they performed classification with multiple classifiers, such as random forest. However, the methods using only a single packet cannot effectively describe the changing behaviors of the resolved IPs. Therefore, most passive methods perform detection based on the packet sets (Almomani, 2018; Truong et al., 2020; Wang and Chen, 2019a). Almomani et al. (Almomani, 2018) made statistics on the traffic of a domain name for one month, extracted 14-dimentional features such as source and destination IPs, and average TTL values, and performed classification based on adaptive evolving fuzzy neural networks.

In addition, some methods use additional databases as supplement, such as geographical locations, ASNs, etc. (Almomani et al., 2021; Al-Nawasrah et al., 2018; Lombardo et al., 2018; Surjanto and Lim, 2020; Kumar and Xu, 2018; Wang and Chen, 2019b) Almomani et al. (Almomani et al., 2021) and Al-Nawasrah (Al-Nawasrah et al., 2018) extracted 7-dimensional features from DNS response packets, such as number of resolved IPs, number of ASNs, and packet size, and perform classification with AdeSNN algorithm they proposed. Lombardo et al. (Lombardo et al., 2018) extracted 6-dimensional static features and 4-dimensional history-based features, including maximum answer length, number of IPs, and number of ASNs. Each dimension of the features was normalized, then they calculated the weighted summation to obtain the Aggregation value *A*, and performed detection according to the threshold.

3 METHOD

3.1 Motivation

Fast-Flux domain names usually correspond to a large number of changing resolved IPs, and these IPs are widely distributed. Hence, Fast-Flux domain names are different from normal domain names in terms of spatial information, such as geographical distributions and network characteristics. Therefore, we perform detection based on spatial features, which are extracted from resolution results we actively obtained. We use the *dig* command in Linux system to obtain the resolution result. *Dig* command provide functions of *A* record request and NS record request. The *A* record (recorded as the "dig A") request is used to obtain the resolved IPs of the domain name, which are stored in the Answer resource records (RRs) in the response packets; while the NS record (recorded as the "dig NS") request is used to obtain the authoritative nameservers (NS) and the IPs of the nameservers (stored in the Additional RRs in DNS response packet). The Answer RRs of "dig A" response packets usually contain CNAME and *A* records (i.e. IP record), in which CNAME record resolves the domain name to another domain name, and *A* record resolves the domain name or CNAME to an IP address. The details of Answer RRs will be described in Section 3.3.

We obtain the resolution results of domain name based on the "dig A" request, and use the CNAME and IP records as nodes to construct the resolution spatial relationship graph (RSRG). Since an important characteristic of Fast-Flux domain name is the changing resolved IPs, and response packet of a single "dig A" request cannot reflect that. Therefore, we request the domain name for multiple times, and construct the RSRG with all response packets of the requests. In addition, Content Distribution Network (CDN) uses a distributed way to build a group of servers on the Internet, and provides services for users through the principles of nearest service, and load balancing, etc. Therefore, a CDN domain name also corresponds to a large number of resolved IPs in the RSRG, and the IPs may be changed, which leads to interference to the detection of Fast-Flux domain names (Chen et al., 2019). For a CDN domain name, the resolved IPs obtained in different locations would be different, and the change frequency of the resolved IPs is relatively low; while for the Fast-Flux domain name, the resolved IPs obtained in different location may be the same, and the resolved IPs may be changed rapidly. Therefore, in order to better distinguish Fast-Flux from CDN domain names, we also execute the ”dig A” request for 5 times from two different locations according to the idea of Chen et al. (Chen et al., 2019). And all IP and CNAME records are taken as nodes to enrich the structural features of RSRG.

The object we process in the detection is a single domain name, that is, an independent Fully Qualified Domain Name (FQDN), such as ”www.myjaasceb.myjaacb.xjkuh.top”. The second-level part of a domain name (such as ”xjkuh.top”) usually represents the owner of the domain name. And for large normal networks, there are usually more nameservers. Therefore, we choose to extract the second-level part of the domain name, and perform an ”dig NS” request for the second-level part synchronously every time we perform a ”dig A” request, to obtain the nameserver information of the second-level part. In addition, the geographical locations (Geo) and autonomous system numbers (ASN) could well reflect the distribution of the IPs, and WHOIS information could provide the creation date of a domain. Therefore, we choose these four categories of data as the detection basis, and our data acquisition process is shown in Figure 1. First, we perform 5 ”dig A” and 5 ”dig NS” requests from each of the 2 locations that is in different cities, and obtain information including CNAME, IP, NS, Additional, etc., and make a one-to-one correspondence between the ”dig A” response packets and the ”dig NS” response packets. After that, *Maxmind* (Maxmind Inc., 2022) is used to acquire geographical locations and ASNs of all IPs obtained from 10 requests in an offline way, and WHOIS request is executed to acquire the creation date of the domain name.

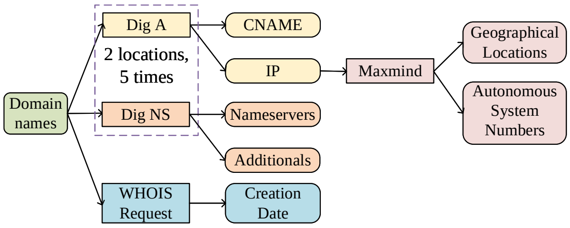


Figure 1: Process of data acquisition.

After the data acquisitions, all the obtained CNAME and IP records in the 10 ”dig A” requests are regarded as nodes in RSRG, and we perform features extraction for each node based on the ”dig A” result, the corresponding ”dig NS” data packets, Geo, ASN, and WHOIS information. Finally, deep learning method is used to extract the spatial features of the domain name from the obtained RSRG. Graph convolutional network (GCN) (Kipf and Welling, 2016) is a commonly used graph classification methods, which updates the node features by the features of their neighbors. Since GCN model is designed for node feature calculation or node classification, in order to apply it to the classification of the whole graph, we refer to the idea of Knyazev et al. (Knyazev et al., 2018) and use Max Pooling algorithm to extract features of the whole graph, named domain resolution spatial features, and conduct classification based on a fully-connected layer.

3.2 Model Structure

We propose a Fast-Flux malicious domain name detection model based on the domain resolution spatial features. The structure of our model is shown in Figure 2. The model mainly contains 3 layers, which are the data acquisition layer, RSRG construction layer, and spatial feature extraction and classification layer.

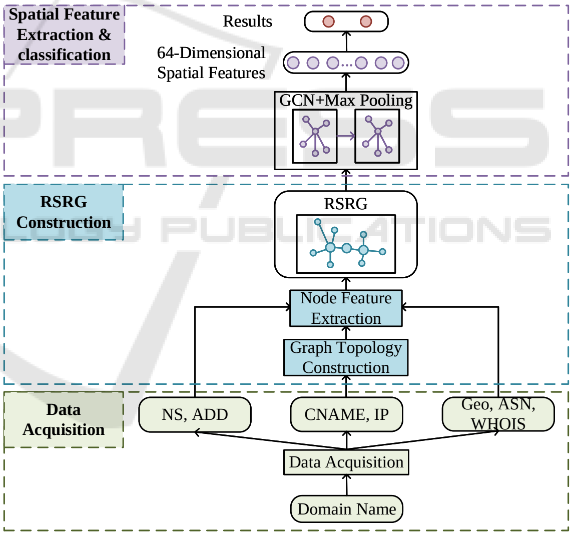


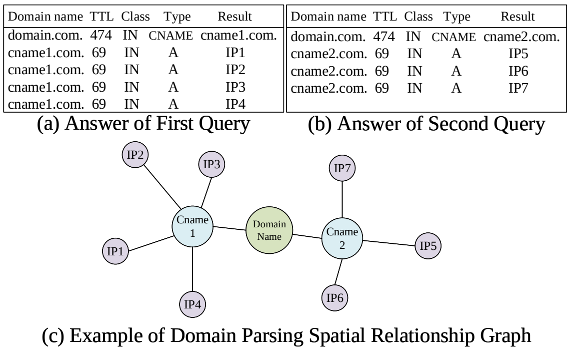
Figure 2: Structure of the Fast-Flux domain name detection method.

In the data acquisition layer, operations including ”dig A”, ”dig NS” and WHOIS are used to obtain the IP resolution results, NS information and WHOIS information of the domain name, and Maxmind (Maxmind Inc., 2022) is used to obtain the geographical locations and ASNs of the resolved IPs, as described in Section 3.1. In RSRG construction layer, we first construct the topology of the RSRG based on the ”dig A” results, which will be introduced in Section 3.3; we then construct a 16-dimensional feature vector for each node in the RSRG from the original results of

”dig A”, ”dig NS” and WHOIS, etc., which will be introduced in Section 3.4. In spatial feature extraction and classification layer, we first adopt multi-layer GCN model and Max Pooling algorithm to extract 64-dimensional spatial features, which will be introduced in Section 3.5; and finally, a fully-connected layer is used to output a 2-dimensional feature vector, which is the classification result of the domain name.

3.3 Topology Structure Construction of RSRG

First, we select CNAME and *A* records from the Answer RRs in ”dig A” results as the CNAME nodes and IP nodes in the RSRG. Each CNAME or IP is associated with the domain name node or the CNAME node, which is in accordance with the relations exist in the Answer RRs. And finally, we obtain the topology of RSRG, which is a divergent tree structure graph, in which the domain name is the central node. As shown in Figure 3, assume that the detected domain name is ”domain.com” and we perform ”dig A” request twice. The Answer RRs is shown in Figure 3(a) and Figure 3(b), which contain 5 and 4 records respectively. Each record contains the domain name, time to live (TTL), class, type, and result. In Figure 3(a), the CNAME node ”cname1.com” depends on the requested domain name ”domain.com”, and the four IP nodes depend on the CNAME node ”cname1.com”. Since the results returned from each ”dig A” request may be different, we choose to retain all the results to build the RSRG. We believe that each correlation between domain name, CNAME and IP nodes has bidirectional influences, and the GCN model we use is based on undirected graph. Therefore, we define the RSRG as an undirected graph. The final RSRG for ”domain.com” we construct is shown in Figure 3(c).



Domain name	TTL	Class	Type	Result
domain.com.	474	IN	CNAME	cname1.com.
cname1.com.	69	IN	A	IP1
cname1.com.	69	IN	A	IP2
cname1.com.	69	IN	A	IP3
cname1.com.	69	IN	A	IP4

(a) Answer of First Query

Domain name	TTL	Class	Type	Result
domain.com.	474	IN	CNAME	cname2.com.
cname2.com.	69	IN	A	IP5
cname2.com.	69	IN	A	IP6
cname2.com.	69	IN	A	IP7

(b) Answer of Second Query

(c) Example of Domain Parsing Spatial Relationship Graph

Figure 3: Example of RSRG topology structure construction.

3.4 Node Feature Extraction in RSRG

In terms of node features, we design 16-dimensional features for each node based on the obtained data, including 5-dimensional IP resolution features, 4-dimensional NS resolution features, 6-dimensional geographical and ASN features, and 1-dimensional WHOIS feature. There are three types of nodes, which are domain name node, CNAME node, and IP node. Different types of nodes have different ways of feature calculation. Each dimension of the node features is finally divided by the maximum value of such dimension in the dataset for normalization, to improve the effect of the features extracted by the GCN model.

(1) IP Resolution Features.

The IP resolution features of nodes are extracted from the ”dig A” response packets, and the 5-dimensional features we extracted are shown in Table 1. Among them, *Times_A1* and *Times_A2* are used to measure the changes of resolved IPs. If such 2-dimensional features of all nodes are 5, it proves that all the records have not changed; on the contrary, the resolved IPs have changed for many times. TTL value indicates the valid time of a resolved CNAME or IP. The resolved IPs of Fast-Flux domain name changes quickly, so the TTL value is relatively small. Since that TTL value changes periodically, the maximum value of the TTL could better reflect the valid time of the result. In addition, Fast-Flux domain name is usually only accessed by infected hosts, it is more likely that the local DNS server cannot find the resolution result and needs to perform iterative query, which results in a longer response time. Moreover, the packet size is also related to the number of results contained in the packet. Therefore, we believe that the response time and packet size have important value for Fast-Flux domain name detection. For a CNAME or IP node, the 5-dimensional features are extracted according to the ”dig A” response packets containing such CNAME or IP resolution result; for the domain name node, they are extracted according to all the 10 ”dig A” response packets.

Table 1: Description of IP resolution features.

Name	Meaning
Times_A1	Times resolved at location 1
Times_A2	Times resolved at location 2
Max_ttl	Maximum of TTL value
Min_A_time	Minimun response time of *A* requests
Max_A_size	Maximun packet size of *A* requests

(2) NS Resolution Features.
The NS resolution features of nodes are extracted from the ”dig NS” response packets of the second-level part of the domain name. The 4-dimensional features we extracted are shown in Table 2. The NS records store the nameservers, while the Additional records stores the IP addresses of the nameservers. The second-level part of the domain name usually represents the owner of the domain name. For a large normal network, there are generally more nameservers, that is, more NS records and more Additional records. Since each ”dig A” response packet is one-to-one corresponded with a ”dig NS” response packet, for a CNAME or IP node, the 4-dimensional features are extracted from the ”dig NS” response packets corresponding to the ”dig A” response packets that contains such CNAME or IP record; for the domain name node, they are extracted according to all the 10 ”dig NS” response packets.

Table 2: Description of NS resolution features.

Name	Meaning
Num_NS	Number of NS records
Num_ADD	Number of additional records
Min_NS_time	Minimun response time of NS requests
Max_NS_size	Maximun packet size of NS requests

(3) Geographical and ASN Features.
The geographical location indicates the location of an IP in the real world, and the ASN indicates the owner of the IP. The 6-dimensional features we extracted are shown in Table 3. FFSN usually controls hosts in other networks as relay servers, such hosts are usually distributed in different locations, resulting in highly dispersed IPs. The number of cities, countries, continents, postcodes, and registered countries of IPs can measure the geographical dispersion of the resolved IPs. In addition, the hosts controlled by FFSN usually belong to different owners, resulting in more distinct ASNs, so we use the number of ASNs to measure the number of IP owners. For an IP node, the 6-dimensional features are all 1; for a CNAME node, they are statistically extracted according to its adjacent IP nodes; for the domain name node, they are statistically extracted according to all IP nodes.

Table 3: Description of geographical and ASN features.

Name	Meaning
Num_city	Number of cities
Num_country	Number of countries
Num_continent	Number of continents
Num_postal	Number of postal numbers
Num_reg_country	Number of register countries
Num_ASN	Number of ASNs

(4) WHOIS Feature.
The Fast-Flux domain name may change sometimes, resulting in shorter registration time; while domain names for normal services are relatively stable, resulting in longer registration time. The creation date of the domain name is included in WHOIS information; we choose to extract such time, and subtract the creation date from the current time to get the *Life_time* of the domain name as one dimension of the node features. If such WHOIS information is not found or creation date is not included in the WHOIS information, the *Life_time* is set to 0. Since we only obtain the WHOIS information of the domain name, the value for all nodes in the RSRG is all set to the same value.

3.5 Spatial Feature Extraction from RSRG

After the construction of the RSRG, we use the multi-layer GCN to extract spatial features from it. Since the GCN model updates features of each node based on its adjacent nodes, in an n-layer GCN network, each node features may be affected by the node features within a distance (i.e., the shortest path in graph) of *n*.

As shown in Figure 3(c), the RSRG we construct contains 2 CNAME nodes and 7 IP nodes. When there is no association between two CNAME nodes, the maximum distance of nodes in the graph is the distance between two IP node that associated with different CNAME nodes, such as distance between *IP1* and *IP5* in Figure 3(c), that is 4. However, in the actual RSRG, there may be associations between two CNAME nodes, which leads to larger distance between nodes in the graph. For instance, in the RSRGs in our dataset, the maximum distance between nodes is 7. For common GCN model used for node classification, the number of layers is usually set to 2 or 3. More GCN layers would lead to poor differentiation of nodes, and the feature vectors of different nodes tend to be consistent. However, our goal is to extract the features of the whole graph, so we do not need to consider the differentiation of node features. Therefore, we choose to use GCN model with 1-10 layers for feature extraction, and select the optimal number of GCN layers as 8 through experiments. Here, the node features output by each GCN layer is set to 64-dimensions, and the Max Pooling algorithm is adopted to select the maximum value of each dimension from all nodes features output by the last GCN layer, and the 64-dimensional features are finally obtained as the spatial features of the RSRG.

4 EXPERIMENTS

4.1 Datasets and Indicators

The dataset we used is self-constructed which is based on the Alexa domain names, CDN domain names, and Fast-Flux domain names collected by ourselves. The Alexa and CDN domain names are regarded as normal domain names and be acquired from Alexa Topsites (Alexa Web Information Company, 2021) and github (Vincent, 2021) respectively. For Fast-Flux domain names, we first obtained malicious domain names from spam Archive (Guenter, 2022), phishstats (Phishstats, 2022), phishtank (Phishtank, 2022), and cert.pl (Cert Polska, 2020). Then, *dig* command was used in Beijing to obtain the resolved IPs from ”114.114.114.114” DNS server for 5 consecutive days, and according to the idea of Nagunwa et al. (Nagunwa et al., 2022), only the domain names with resolved IPs and the resolved IPs have changed at least one time were retained as the final Fast-Flux domain names. Finally, 9,674 Alexa domain names, 8,638 CDN domain names and 8,709 Fast-Flux domain names were selected as the final dataset of this paper.

For all the domain names in the dataset, we used the ”dig A” command to obtain the resolution results of the domain names for 5 consecutive days from Beijing and Nanjing respectively, and used the ”dig NS” command to obtain the NS records of second-level part of each domain name. Then an offline database named MaxMind (Maxmind Inc., 2022) is used to acquire the geographical and ASN information for each resolved IPs, and WHOIS information of the domain name is acquired.

Both CDN and Fast-Flux domain names correspond to a large number of resolved IPs, which may affect the effectiveness of the detection model. To better evaluate our detection method, we built three sub-datasets based on the full dataset, which are Full dataset (Alexa + CDN + Fast-Flux), Alexa_FF dataset (Alexa + Fast-Flux), and CDN_FF dataset (CDN + Fast-Flux). Finally, we used the average results of 5-fold cross-validation as the final testing results. For all the three sub-datasets, 4 indicators including accuracy (Acc), precision (Pre), recall (Rec) and F1 value are used.

4.2 Experimental Settings

In this paper, two active detection methods with best performance are selected for comparison, which are FCDR (Chen et al., 2019) and FluxOR (Passerini et al., 2008). In addition, we set up multiple ablation experiments to verify the effectiveness of the virous designs of our method.

We conducted 5 groups of experiments in total.

- **Experiment 1**, comparison experiment. We compared the effects of our method with FCDR and FluxOR on the three sub-datasets.
- **Experiment 2**, effectiveness experiment for node features. Taking the IP resolution features as the base, we successively added NS resolution features, Geographical and ASN features and WHOIS features, to prove the effectiveness of each set of node features.
- **Experiment 3**, effectiveness experiment for data acquisition methods. First, based on acquisition in 2 locations, we compared the detection effects of model when using data of 1-5 days, to show the impact of the data acquisition times on the performance. Then, based on acquisition for 5 times, we compared the detection effects of model when acquiring data in 1 or 2 locations, to show the effectiveness of data acquisition in 2 locations.
- **Experiment 4**, effectiveness experiment for node feature preprocessing. We compared the detection effects of non-normalization of node features, normalization, and adding an additional fully-connected layer, and selected the best node feature preprocessing method.
- **Experiment 5**, effectiveness experiment for number of GCN layers. We compared the detection effects of GCNs with number of layers ranging from 1 to 10 respectively, to select the optimal number of GCN layers.

4.3 Results

4.3.1 Comparison Experiment

In this experiment, our method is compared with two existing methods, which are FCDR (Chen et al., 2019) and FluxOR (Passerini et al., 2008). FCDR (Chen et al., 2019) acquired resolution data for 5 times from 2 DNS server, which is the same as our method. They extracted 2 CNAME and 4 IP records from the 2 resolution results of each time to construct 102-dimensional features, and finally obtained a feature sequence with length of 5, and then performed classification using LSTM model. In this method, the 102-dimensional features include 6-dimensional features to indicate whether the CNAMEs and IPs are in the whitelist. Since we did not construct the whitelist, we consider two ways to deal with such 6-dimensional features: directly remove them to form 96-dimensional features, or set all of them to 0. By comparing the experimental results, we found that the overall effect is relatively better when setting all 6-dimensional features to 0, so such result is taken as the final effect of the method. FluxOR (Passerini et al., 2008) extracted 9-dimensional features for each domain name, including domain name registration features, network availability features, and agent heterogeneity features, and used Naive Bayes Classifier to perform classification. Our method iterates 50 epochs in the training process without setting dropout, and the initial learning rate is set to 0.001, and is decayed to 0.1 of the current learning rates after the 25th and 35th epochs. The results of each comparison method are shown in Table 4.

Table 4: Performance of comparison experiments.

Dataset	Indicator	FCDR	FluxOR	Ours
Full	Acc	90.30%	66.39%	94.98%
	Pre	86.12%	48.67%	94.16%
	Rec	83.25%	78.52%	89.38%
	F1	84.66%	60.09%	92.02%
Alexa _FF	Acc	87.95%	60.86%	94.26%
	Pre	85.72%	55.97%	94.73%
	Rec	88.15%	81.50%	93.05%
	F1	86.91%	66.36%	93.88%
CDN _FF	Acc	93.34%	84.22%	95.24%
	Pre	92.84%	79.84%	95.90%
	Rec	93.99%	91.73%	94.56%
	F1	93.41%	85.37%	95.22%

From the results, our method achieved significantly better effect than the existing two methods, when comparing on the three sub-datasets using 5-fold cross-validation. For FluxOR method, the features they constructed are relatively simple, which achieved poor detection effect. The FCDR method only considered 1 CNAME and 2 IP records in each resolution result, such features ignored the spatial features of the resolved IPs, such as the number and distribution of the resolved IPs. And the overall effect is also lower than our method.

4.3.2 Effectiveness Experiment for Node Features

The node features in RSRG are mainly divided into four sets: IP resolution features (marked as ”IP”), NS resolution features (marked as ”NS”), Geographical and ASN features (marked as ”Geo_ASN”) and WHOIS features. In this experiment, each method constructed RSRG based on the data of all the 5-days, and used an 8-layer GCN to extract spatial features and perform classification. The experiment results are shown in Table 5.

From the results, when only using IP resolution features, our model reached accuracies of more than 90% on the three sub-datasets, which proves the effectiveness of the IP resolution features. After adding NS resolution features, the effects on three sub-datasets are significantly improved, indicating that the NS resolution features are of great value in distinguishing between Fast-Flux and normal domain names. After adding Geographical and ASN features, the effect on Alexa_FF dataset changed little, but the effect on CDN_FF dataset was significantly improved, indicating that such features are of great value in distinguishing between Fast-Flux and CDN domain names. Finally, after adding WHOIS features, the effects of the three sub-datasets are significantly improved, which proves the effectiveness of WHOIS features.

4.3.3 Effectiveness Experiment for Data Acquisition Methods

This paper performs resolutions for 5 times in 2 locations, and obtains 10 times of resolution data to construct RSRG for Fast-Flux domain name detection. Among them, acquisition from 2 locations is mainly used to distinguish between Fast-Flux and CDN domain names, and acquisition for 5 times is mainly used to measure the changes of the resolved IPs. This experiment was used to compare the detection effects of our spatial features with data of different request times. In this experiment, each method constructed RSRG with 16-dimensional node features, and used an 8-layer GCN to extract spatial features and perform classification. The experiment results are shown in Table 6.

Comparing the result of data acquisition in 2 locations for 1-5 days, the detection effect of the model was on the rise with the increase of days. Moreover, it is found that the difference of detection effects between data of 1-day and 2-day is significantly higher than that between other contiguous pairs. After analysis, we believe that using data of 1-day, i.e. only one group of resolution result is reserved, the changing behavior of IPs cannot be described. However, when using data of 2-days, there are two groups of resolution results, changes of IPs can be effectively discovered. Therefore, the detection effect is greatly improved. When continuously increasing the days of data, it would be better to describe the changes of IPs, but the increase was relatively small compared to the increase of 1 to 2 days.

Comparing the result of data acquisition for 5 days in 1 and 2 locations, it shows that for Alexa_FF dataset, the detection effects of the two configurations are similar, indicating that it has no significant impact on the dataset when acquiring data from 2 locations.

Table 5: Performance of different node features.

Dataset	Indicator	IP	IP + NS	IP +NS + Geo_ASN	IP +NS + Geo_ASN + WHOIS
Full	Acc	92.13%	93.56%	93.59%	94.98%
	Pre	91.04%	94.15%	94.40%	94.16%
	Rec	83.83%	85.34%	85.16%	89.38%
	F1	87.28%	89.52%	89.55%	92.02%
Alexa_FF	Acc	92.10%	93.43%	93.18%	94.26%
	Pre	92.01%	94.57%	94.28%	94.73%
	Rec	91.24%	91.38%	91.14%	93.05%
	F1	91.62%	92.94%	92.68%	93.88%
CDN_FF	Acc	90.46%	91.70%	92.92%	95.24%
	Pre	92.18%	92.60%	92.15%	95.90%
	Rec	88.53%	90.78%	93.90%	94.56%
	F1	90.30%	91.66%	93.02%	95.22%

Table 6: Performance of different data acquisition methods.

Dataset	Indicator	2 locations 1 day	2 locations 2 days	2 locations 3 days	2 locations 4 days	2 locations 5 days	1 location 5 days
Full	Acc	93.64%	94.58%	94.80%	94.97%	94.98%	94.90%
	Pre	93.44%	93.74%	93.83%	94.44%	94.16%	94.52%
	Rec	86.34%	89.15%	89.76%	89.69%	89.38%	89.38%
	F1	89.74%	91.38%	91.75%	92.00%	92.02%	91.87%
Alexa_FF	Acc	93.03%	93.88%	94.07%	94.14%	94.26%	94.32%
	Pre	94.29%	94.54%	94.85%	94.73%	94.73%	94.27%
	Rec	90.79%	92.42%	92.50%	92.78%	93.05%	93.71%
	F1	92.50%	93.47%	93.66%	93.74%	93.88%	93.98%
CDN_FF	Acc	93.88%	94.73%	95.07%	94.94%	95.24%	94.70%
	Pre	94.43%	95.50%	96.11%	95.71%	95.90%	95.73%
	Rec	93.33%	93.93%	93.99%	94.16%	94.56%	93.62%
	F1	93.87%	94.70%	95.04%	94.92%	95.22%	94.66%

For CDN_FF dataset and Full dataset, it shows that when using data of 2 locations, the detection effect is significantly higher than that of a single location. Therefore, it is proved that obtaining data from two locations is of great value in distinguishing between Fast-Flux and CDN domain names.

4.3.4 Effectiveness Experiment for Node Feature Preprocessing

Since we extract 16-dimensional features for each node in the RSRG, the value distributions of different dimensions are quite different, hence we choose to divide each dimension of features by its maximum value for normalization. In addition, we consider adding a fully-connected layer after normalization to transform the node features. Here we compared the detection effects before and after normalization of node features, and the detection effects before and after adding an additional fully-connected layer. In this experiment, each method constructed RSRG with 16-dimensional node features based on the data of all 5-days, and used an 8-layer GCN to extract spatial features and perform classification. The experiment results are shown in Table 7.

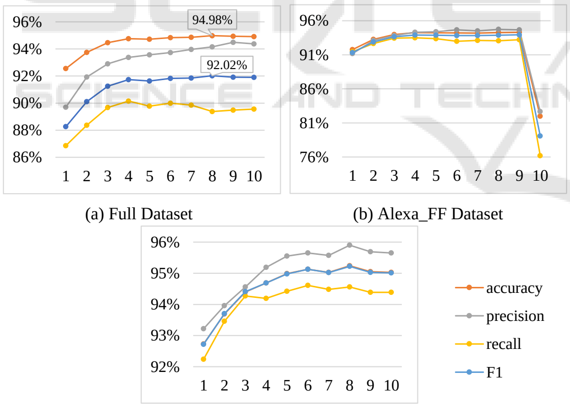
Table 7: Performance of different node features processing methods.

Dataset	Indicator	Non-Normal	Normal	Normal + FC
Full	Acc	92.88%	94.98%	71.40%
	Pre	93.90%	94.16%	61.88%
	Rec	83.33%	89.38%	32.09%
	F1	88.30%	92.02%	41.96%
Alexa_FF	Acc	91.02%	94.26%	63.50%
	Pre	93.64%	94.73%	64.59%
	Rec	86.97%	93.05%	50.70%
	F1	90.18%	93.88%	56.79%
CDN_FF	Acc	93.38%	95.24%	77.18%
	Pre	95.43%	95.90%	70.59%
	Rec	91.19%	94.56%	94.26%
	F1	93.26%	95.22%	80.61%

From the result, after normalization of node features, the detection effect of the model has significantly improved. However, after adding the additional fully-connected layer, the detection effect decreased significantly. We believe that the 8-layer GCN already has a high complexity, the additional fully-connected layer further increases the complexity of the model, resulting in the risk of overfitting or difficulty in convergence, which leads to the decrease of the detection effect. Therefore, we finally choose to only perform normalization for node features of RSRG.

4.3.5 Effectiveness Experiment for Number of GCN Layers

This experiment compares the effects of GCN models with different layers on the effectiveness of the extracted spatial features. To select the optimal number of GCN layers for spatial feature extraction, we tested GCNs with layers ranging from 1 to 10 respectively. In this experiment, each method constructed RSRG with 16-dimensional node features based on the data of all 5-days, and used different GCNs to extract spatial features and perform classification. The experiment results are shown in Figure 4. Figure 4(a), (b), and (c) respectively shows the change trend of each indicator on Full, Alexa_FF and CDN_FF datasets, when number of GCN layers range from 1 to 10.



(a) Full Dataset

(b) Alexa_FF Dataset

(c) CDN_FF Dataset

Figure 4: Performance of different number of GCN layers.

From the results, with the increasing number of GCN layers, the detection effect shows an increasing trend on the whole, which proves that, GCN model with more layers is helpful to improve the detection effect. We believe that more GCN layers could effectively fuse the features of nodes with farther distances, and achieved deeper abstraction level of the extracted feature, which result in higher efficiency of the model.

However, with the number of GCN layers increasing, the improvement of the detection effect shows a gradually decreasing trend in the three sub-datasets. And the overall detection effect is the best when the 8-layer GCN is used. Moreover, when using the 10-layer GCN, the model did not converge for twice in the 5-fold cross-validation on the Alexa_FF dataset, resulting in an average accuracy of only 82%, which is much lower than that of GCNs with 1 to 9 layers. We believe that the maximum distance between nodes in the current RSRGs is 7, and more GCN layers could not integrate features of more nodes. In addition, the GCN model with more layers could improve the abstraction ability of features; however, it is more likely for such model to encounter problems of overfitting and non-convergence, resulting in the decrease of the detection effect. Therefore, we choose the 8-layer GCN as the final model.

5 CONCLUSIONS

In this paper, a Fast-Flux malicious domain name detection method using domain resolution spatial features is proposed. The method obtains the resolution information through *dig* command in Linux system, and constructs the topology of the resolution spatial relationship graph (RSRG) based on the CNAME and IP records. Then the method extracts 16-dimensional node features, including IP resolution features, NS resolution features, geographical and ASN features, and WHOIS feature. Finally, the GCN model combined with Max Pooling algorithm is used to extract spatial features from the RSRG and perform classification. The paper constructs a dataset based on 4 malicious domain name websites, Alexa domain names and CDN domain names, and perform testing based on the dataset. The experimental result shows that our method achieves an accuracy of 94.98%and an F1 value of 92.02%, which is higher than that of the existing similar methods.

At present, our method only considers the spatial features of the domain names. We will consider mining more relevant information from existing data, such as text features in WHOIS information, to further improve the detection effect of the model. And then, we will deploy the detection model in actual network to detect Fast-Flux domain names and discover the attacks.

REFERENCES

Al-Duwairi, B., Al-Hammouri, A., Aldwairi, M., and Paxson, V. (2015). Gflux: A google-based system for fast flux detection. In *2015 IEEE Conference on Com-*

munications and Network Security (CNS), pages 755–756. IEEE.

Al-Duwairi, B., Jarrah, M., and Shatnawi, A. S. (2021). Passvm: A highly accurate fast flux detection system. *Computers & Security*, 110:102431.

Al-Nawasrah, A., Al-Momani, A., Meziane, F., and Alauthman, M. (2018). Fast flux botnet detection framework using adaptive dynamic evolving spiking neural network algorithm. In *2018 9th International Conference on Information and Communication Systems (ICICS)*, pages 7–11. IEEE.

Alexa Web Information Company (2021). Topsites. https://www.alexa.com/topsites.

Almomani, A. (2018). Fast-flux hunter: a system for filtering online fast-flux botnet. *Neural Computing and Applications*, 29(7):483–493.

Almomani, A., Al-Nawasrah, A., Alauthman, M., Al-Betar, M. A., and Meziane, F. (2021). Botnet detection used fast-flux technique, based on adaptive dynamic evolving spiking neural network algorithm. *International Journal of Ad Hoc and Ubiquitous Computing*, 36(1):50–65.

Berger, A., D'Alconzo, A., Gansterer, W. N., and Pescapé, A. (2016). Mining agile dns traffic using graph analysis for cybercrime detection. *Computer Networks*, 100:28–44.

Cafuta, D., Sruk, V., and Dodig, I. (2018). Fast-flux botnet detection based on traffic response and search engines credit worthiness. *Tehnički vjesnik*, 25(2):390–400.

Cert Polska (2020). List of malicious domains. https://cert.pl/en/posts/2020/03/malicious_domains/.

Chandavarkar, B. et al. (2018). Maldetect: A framework to detect fast flux domains. In *2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pages 141–146. IEEE.

Chen, X., Li, G., Zhang, Y., Wu, X., and Tian, C. (2019). A deep learning based fast-flux and cdn domain names recognition method. In *Proceedings of the 2019 2nd International Conference on Information Science and Systems*, pages 54–59.

Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. (2015). A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 542–553.

Guenter, B. (2022). Spam archive. http://untroubled.org/spam/.

Holz, T., Gorecki, C., Rieck, K., and Freiling, F. C. (2008). Measuring and detecting fast-flux service networks. In *Ndss*.

Hsu, F.-H., Wang, C.-S., Hsu, C.-H., Tso, C.-K., Chen, L.-H., and Lin, S.-H. (2014). Detect fast-flux domains through response time differences. *IEEE Journal on Selected Areas in Communications*, 32(10):1947–1956.

Huang, S.-Y., Mao, C.-H., and Lee, H.-M. (2010). Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection. In *Proceedings of the 5th ACM symposium on information, computer and communications security*, pages 101–111.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Knyazev, B., Lin, X., Amer, M. R., and Taylor, G. W. (2018). Spectral multigraph networks for discovering and fusing relationships in molecules. *arXiv preprint arXiv:1811.09595*.

Kumar, S. A. and Xu, B. (2018). A machine learning based approach to detect malicious fast flux networks. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1676–1683. IEEE.

Lin, H.-T., Lin, Y.-Y., and Chiang, J.-W. (2013). Genetic-based real-time fast-flux service networks detection. *Computer Networks*, 57(2):501–513.

Lombardo, P., Saeli, S., Bisio, F., Bernardi, D., and Massa, D. (2018). Fast flux service network detection via data mining on passive dns traffic. In *International Conference on Information Security*, pages 463–480. Springer.

Maxmind Inc. (2022). Geoip databases & services: Industry leading ip intelligence. https://www.maxmind.com/en/geoip2-services-and-databases.

Nagunwa, T., Kearney, P., and Fouad, S. (2022). A machine learning approach for detecting fast flux phishing hostnames. *Journal of Information Security and Applications*, 65:103125.

Passerini, E., Paleari, R., Martignoni, L., and Bruschi, D. (2008). Fluxor: Detecting and monitoring fast-flux service networks. In *International conference on detection of intrusions and malware, and vulnerability assessment*, pages 186–206. Springer.

Phishstats (2022). Phishstats. https://phishstats.info/.

Phishtank (2022). Phishtank. https://www.phishtank.com/.

Rana, S. and Aksoy, A. (2021). Automated fast-flux detection using machine learning and genetic algorithms. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6. IEEE.

Surjanto, W. and Lim, C. (2020). Finding fast flux traffic in dns haystack. In *International Conference on Critical Information Infrastructures Security*, pages 69–82. Springer.

Truong, D.-T., Tran, D.-T., and Huynh, B. (2020). Detecting malicious fast-flux domains using feature-based classification techniques j. *J Internet Technol.*, 21:1061–72.

Vincent, Y. (2021). Domainfrontinglists. https://github.com/vysecurity/DomainFrontingLists.

Wang, J. and Chen, Y. (2019a). Detection method of fast flux service network based on decision tree algorithm. In *Journal of Physics: Conference Series*, volume 1325, page 012112. IOP Publishing.

Wang, J. and Chen, Y. (2019b). Fast-flux detection method based on dns attribute. In *Journal of Physics: Conference Series*, volume 1325, page 012049. IOP Publishing.

Zhauniarovich, Y., Khalil, I., Yu, T., and Dacier, M. (2018). A survey on malicious domains detection through dns data analysis. *ACM Computing Surveys (CSUR)*, 51(4):1–36.