Fast-flucos：基于 DNS 流量的 Fast-flux 恶意域名检测方法

韩春雨[1,2]，张永铮[2,3]，张玉[1]

（1. 南开大学计算机学院，天津 300071；2. 中国科学院信息工程研究所，北京 100093；
3. 中国科学院大学网络空间安全学院，北京 100049）

摘 要：现有的 Fast-flux 域名检测方法在稳定性、针对性和流量普适性方面存在一些不足，为此提出一种基于 DNS 流量的检测方法 Fast-flucos。首先，采用流量异常过滤和关联匹配算法，以提高检测的稳定性；然后，引入量化的地理广度、国家向量表和时间向量表特征，以加强对 Fast-flux 域名检测的针对性；最后，采用更合理的正负样本和包括深度学习在内的多种机器学习方法确定最佳分类器和最优特征组合，以尽量确保对真实 DNS 流量的普适性。基于真实 DNS 流量的实验表明，Fast-flucos 的召回率、精确率和 ROC_AUC 分别达到了 0.998 6、0.976 7 和 0.992 9，均优于当前主流的 EXPOSURE、GRADE 和 AAGD 等检测方法。

关键词：Fast-flux；域名系统；域名检测；机器学习；深度学习

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2020094

Fast-flucos: malicious domain name detection method for Fast-flux based on DNS traffic

HAN Chunyu[1,2], ZHANG Yongzheng[2,3], ZHANG Yu[1]

1. College of Computer Science, Nankai University, Tianjin 300071, China
2. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
3. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: There are three weaknesses in previous Fast-flux domain name detection method on the aspects of stability, targeting, and applicability to common real-world DNS traffic environment. For this, a method based on DNS traffic, called Fast-flucos was proposed. Firstly, the traffic anomaly filtering and association matching algorithms were used for improving detection stability. Secondly, the features, quantified geographical width, country list, and time list, were applied for better targeting Fast-flux domains. Lastly, the feature extraction were finished by the more suitable samples for trying to adapt to common real-world DNS traffic. Several machine learning algorithms including deep learning are tried for determining the best classifier and feature combination. The experimental result based on real-world DNS traffic shows that Fast-flucos' recall rate is 0.998 6, precision is 0.976 7, and ROC_AUC is 0.992 9, which are all better than the current main stream approaches, such as EXPOSURE, GRADE and AAGD.

Key words: Fast-flux, domain name system, domain name detection, machine learning, deep learning

收稿日期：2019−10−31；**修回日期**：2020−04−22

基金项目：国家自然科学基金资助项目（No.U1736218）；北京市科学技术委员会基金资助项目（No.Z191100007119005）

Foundation Items: The National Natural Science Foundation of China (No.U1736218), Beijing Municipal Science & Technology Commission Project (No.Z191100007119005)

1 引言

随着网络的应用越来越广泛，域名系统（DNS, domain name system）逐渐成为最重要的互联网基础设施之一，许多网络上的基础服务都与其息息相关。与此同时，恶意攻击行为也伴随互联网的快速发展而不断变化。僵尸网络作为当下网络环境中的最大威胁之一，能够迅速分发并扩散大量的蠕虫、木马等计算机病毒进行恶意代码（僵尸程序）的传播[1]。起初的恶意代码往往在程序中预置某一个域名指向其命令与控制服务器（C&C, command&control server），随后就有黑客采用域名生成算法（DGA, domain generation algorithm）技术产生大量的无关域名来隐藏真正的 C&C 域名来躲避拦截。但使用的 IP 地址仍然固定，无法防止因 IP 地址被封堵而直接失效。所以攻击者又转而使用 Fast-flux 技术，它是一种通过不断变更僵尸代理主机来隐藏恶意程序分发点并可实现负载均衡的 DNS 技术。Fast-flux 网络为攻击者的恶意行为提供了高可用性和动态性[2]，其目前被人们熟知的明显特点总结如下[3-5]。

1) 单域名映射的 IP 地址量较大，且在僵尸代理主机迁移阶段 IP 地址会逐渐增多。

2) IP 地址所在地理位置（国家或地区）分布较广。

3) IP 地址变化频度较高。

4) 通常权威映射记录生存空间（TTL, time to live）值较短。

5) 僵尸代理主机所分布的网段数目较多。

6) 僵尸代理主机所在网络所属的组织机构数目较多。

然而，以上特点并非是 Fast-flux 恶意域名所独有的。以下 2 种域名通常也会具有上述特征。

1) 域名循环系统（RRDNS, round-robin domain name system）。RRDNS 是一种典型的用于负载分配、实现负载均衡和高容错率的 DNS 技术，它对用户发出 DNS 请求的响应不是单条 A 记录，而是一个 A 记录列表，这就意味着这种域名也会对应很多服务主机的 IP 地址。RRDNS 域名的 A 记录列表以一种循环的方式来响应连续的 DNS 请求。因此，一系列向 RRDNS 发出的请求会直接得到不同 IP 地址的服务器响应，从而有效地实现了负载均衡。

2) 内容分发网络（CDN, content delivery network）。CDN 是另一种可以实现负载均衡的服务系统，该系统在一定区域内的每个节点都向适合的用户提供完全相同的响应数据。正确设计和实施的 CDN 可以提高访问远程数据的效率，增加带宽和冗余，同时减少时延。因此，该服务响应同一个域名的 DNS 请求时，会返回大量带有不同 IP 地址的 A 记录。CDN 也使用较低的 TTL 值，以便在连接属性等相关参数发生改变时快速做出相应调整。

一直以来，利用 Fast-flux 技术形成的僵尸网络依然广泛存在，由于 CDN 和 RRDNS 的技术特点均与 Fast-flux 相似，因此当二者对应的域名部署在 DNS 上时，可能会被误检为 Fast-flux 恶意域名，这无疑增加了 Fast-flux 域名检测工作的难度[6]。而后出现的 Double-flux 技术[7]使检测工作变得更加困难，该技术是对以前 Single-flux 技术的升级，其由多个 flux 代理主机担任域名解析代理，并由后端的 Mothership 执行域名解析。通过 flux 代理主机的快速切换，避免了以往单个被攻陷的 DNS 服务器被封堵而无法继续提供服务的情况。即使某个 flux 代理主机被列入黑名单，也可通过新的 flux 代理主机继续提供服务。这两项技术的原理分别如图 1 和图 2 所示。

从图 1 和图 2 可以看出，Double-flux 型 Fast-flux 网络在结构上更加复杂，且因单点被封堵而引发故障的可能性也被大大降低。根据僵尸网络的行为过程，无论应对哪种 Fast-flux 技术，高效地检测 Fast-flux 域名都具有重大意义。

2 相关工作

在 Fast-flux 僵尸网络刚出现时，Nazario 等[8]便开始对其展开了研究，分别从 Fast-flux 僵尸网络的域名特性、域名存活时间、网络成员信息、网络大小、网络重合度等方面分析了其行为。Caglayan 等[9]对垃圾邮件、恶意代码和钓鱼这 3 种 Fast-flux 僵尸网络进行了全面的分类跟踪分析。Hu 等[10]通过一个部署在 4 个洲共计 240 个节点的轻量级 DNS 分析 DIGGER，进一步分析了 Fast-flux 网络及其他与其特性相似的网络的 IP 地址特性。Passerini 等[11]采用主动探测的方法以获得比被动接收 DNS 流量更多的 Fast-flux 网络相关信息。通过这些研究工作，人们得到了一系列较成熟的 Fast-flux 域名相关特性。于是，各种检测方法开始相继出现。

文献[12]通过域名对应 IP 地址集合的相似性，对域名进行有监督的分层聚类分析，并采用决策树算法，初步实现了对 Fast-flux 域名的检测。

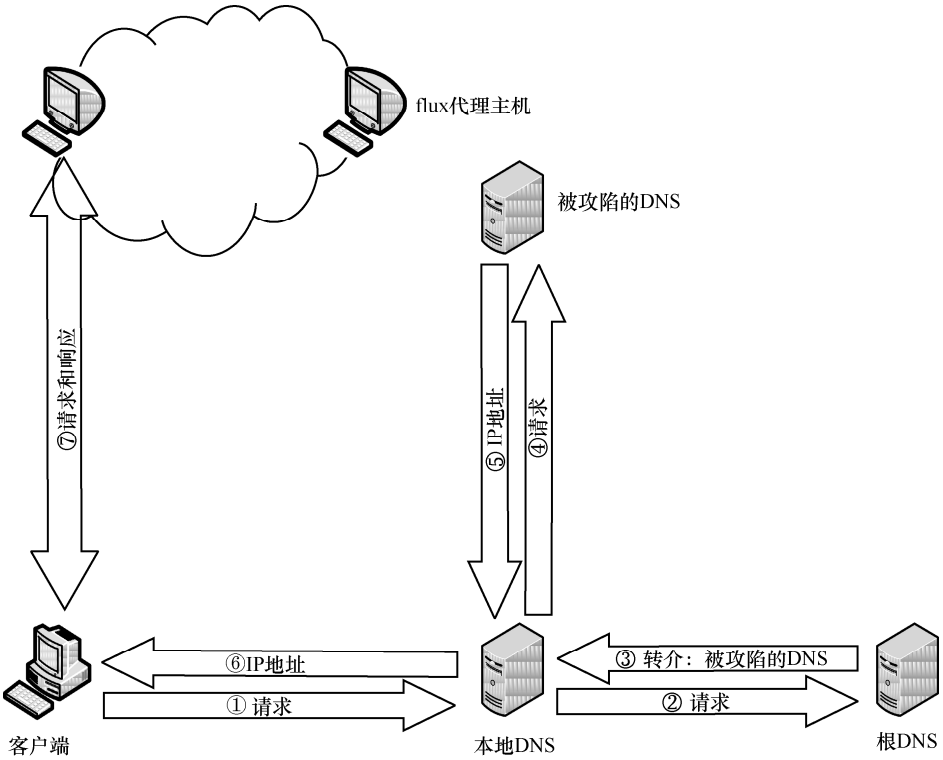

图 1　Single-flux 原理示意

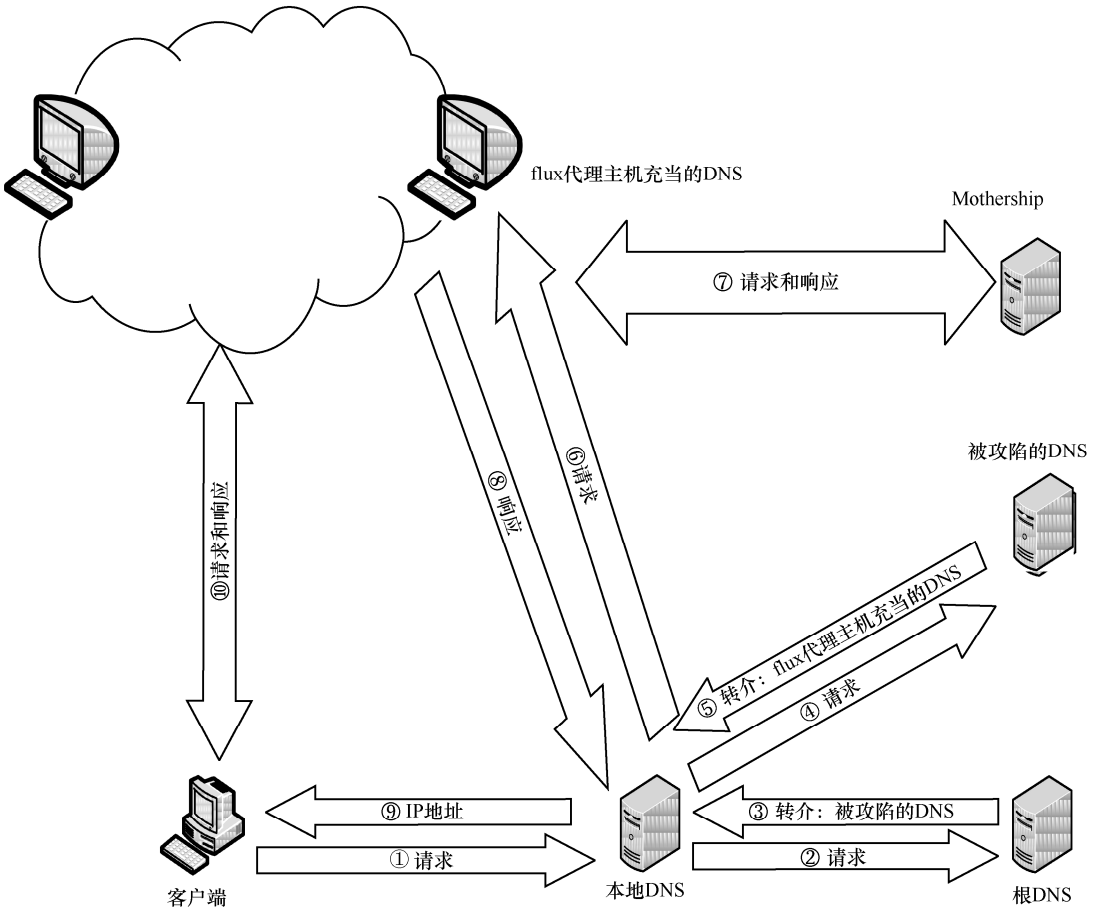

图 2　Double-flux 原理示意

文献[13]提出了一种基于真实编码遗传算法的 Fast-flux 域名检测方法 GRADE（genetic-based real-time detection system），其在文中实验的检测结果可以达到较高的准确率，但主要利用的新特征依赖于递归请求往返时间标准差，而这一参数受网络环境波动的影响较大，直接影响了其性能的稳定性。

文献[14]提出了名为“SSFD（spatial snapshot Fast-flux detection system）”的检测方法，其核心思想是基于空间关系检测 Fast-flux 域名。该方法引入了时区熵这一新特征，并且需要利用“空间快照”技术将样本域名解析列表中每个 IP 地址归属的地理位置都标记在谷歌地图上。这使该方法对所使用的数据库要求较高，必须先创建、维护并每晚更新一个 IP 地址与归属地理位置相对应的数据库，而一旦某域名的某些解析值在库中找不到对应，又面临“值缺失问题”，这势必影响其检测效果的稳定性。

文献[15]基于 3 种可用于区分 Fast-flux 网络和正常网络的参数计算出一个线性分数指标，称为 Flux-score，用以判别 Fast-flux 域名。

文献[16]分析了当时已有的 Fast-flux 网络检测技术，认为检测的关键在于根据网络拓扑结构选取合适的探测位置部署分布式的检测系统，来实现对 Fast-flux 网络的协同检测与关联分析。

文献[17]同样基于访问 Fast-flux 域名时的响应时间每次都会有较大差异的特点，提出了一个通过计算 FF-Score 来判定 Fast-flux 域名的方法。然而该响应时间的差异性特征同样可能受到网络波动的干扰，从而影响检测效果的稳定性。

文献[18]基于决策树算法提出了 EXPOSURE 检测方法，提取了分别与时间、DNS 应答、TTL、命名相关的 4 组特征。其中仅有少部分特征是针对 Fast-flux 域名的特点而提取的，所以该方法在整体上可能仍然欠缺对检测目标的针对性。

文献[19]提出了一种聚类和有监督学习相结合的恶意域名检测思路，强调了如何有效区分 CDN 域名与 Fast-flux 域名，并提取了与之相关的特征，实验达到了较高的准确率。不过该方法训练的 SVM（support vector machine）分类器所用的数据来源于教育网，可能无法稳定有效地适用于常规互联网环境下的 DNS 流量。

通过对上述文献的分析发现，现有方法可能存在如下不足。

1) 受网络环境波动影响较大，导致检测结果的稳定性较差。

2) 缺乏对 Fast-flux 域名的针对性，导致检测结果的召回率或精确率不高。

3) 用于分类器训练的 DNS 流量不具有普适性。

通过分析真实 DNS 流量发现，Fast-flux 域名在大部分时间极少被访问，而在特定时间被集中访问。这项明显区别于 CDN 等正常域名的重要特征一直没得到有效利用。以这项发现为出发点，并结合大洲覆盖、国家或地区距离、国家或地区向量等特征，提出了一种基于 DNS 流量的 Fast-flux 恶意域名检测方法，命名为 Fast-flucos（fast-flux detection based on countries）。

3 方法介绍

Fast-flucos 包含 4 个模块：预处理、特征提取、分类器和关联匹配。预处理模块先从输入的大量 DNS 流量中筛除一些乱码型域名请求记录，再结合所发现的时间分布特性过滤出很多不可能为 Fast-flux 域名的请求记录，从而得到有效的 DNS 流量。特征提取模块根据我国广东省一段时间内 DNS 流量的相应字段并结合已制作的国家或地区距离表计算出所有域名样本的 7 组特征。通过已知标签样本集和计算好的相关特征，分别使用各种可能的特征组合，并结合多种机器学习算法来训练最佳分类器，从而输出一部分 Fast-flux 域名。关联匹配模块利用 WHOIS 注册者信息并结合特定的字符串匹配算法，根据分类器的输出发现更多 Fast-flux 域名。Fast-flucos 的结构如图 3 所示。

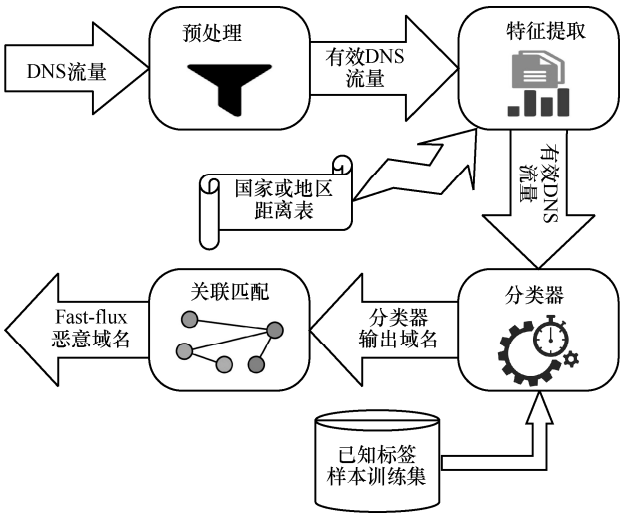

图 3 Fast-flucos 结构

图 3 的 DNS 流量具体是指在特定时间段内，某地区全部由客户端向 DNS 递归服务器发出的 DNS 解析请求的往返信息记录，按一定的采样率组成的 DNS 解析记录列表。每行第一列至最后一列的字段分别表示请求源 IP 地址、请求目的 IP 地址、DNS ID、域名、请求类型、请求分组重复次数、该时间间隔下分组采样率、匹配标识、递归请求标识、权威请求标识、首个应答记录（RR，response record）的 TTL 值、当前获得分组的时间、节点编码、请求分组长度、应答分组长度、首个 RR 的类型、服务器响应标识、省份（或省级地区）标识、通信运营商标识和首个 RR 解析值（含解析 IP 地址列表）。

实验中所使用的数据主要是从我国某主流通信运营商在广东省的 DNS 流量分析所得，时间段是从 2017 年 12 月 1 日—7 日。选取这一段 DNS 流量的主要原因是，使用从微步在线等第三方平台上获取的样本筛查后发现，在这大约 60 万个不重复的符合命名规范的域名中，共出现了 3 022 个与已知标签的 Fast-flux 域名具有相同二级域名的样本，这很利于域名特征的提取。还有一部分数据是直接使用的微步在线提供的域名解析历史记录。

3.1　预处理

通过实验证实所发现的 Fast-flux 域名的重要特性——大部分时间极少被访问，特定时间集中被访问。先将实验所用的时间段内的数据按“域名”字段进行去重处理，得到的域名集合定义为 First 集合。之后的步骤如下[20]。

1) 将该时间段每天的 DNS 流量都以 T min 为单元进行分片。

2) 从 First 集合中选出一个域名定义为 First 域名。

3) 再从原 DNS 流量中该 First 域名出现次数最多的一天中找出其对应的 Second 请求记录。

4)计算该天内每个时间分片的 count_value（计数值，表示 Second 请求记录在该时间分片中出现的总次数）。

某 First 域名的 Second 请求记录，是满足以下任意条件的 DNS 请求记录。

1) “域名”字段与 First 域名完全相同。

2) 与 First 域名的“首个 RR 解析值”字段的交集不为空。

3)“域名”字段与 First 域名具有相同的三级域名。

假设 count_value$_i$（i=1,2,3,…）表示第 i 个分片（window[i]）的 count_value，则根据全部的 count_value$_i$ 可以求出平均值，记为 count_value$_V$，依此可计算出每个域名对应的 AX 值，AX 值为所有 $\frac{\text{count\_value}_i}{\text{count\_value}_V}$ 中的最大值。上述过程可以总结为如下可以得到所有域名 AX 值的算法。

算法 1　AX 赋值算法

输入　First 域名，已分片的 DNS 流量 window[]

输出　AX 值

1) count_value[]={0} //各分片 count_value 初始化

2) for i=0; i<length(window[]); i++　//总分片个数

3)　for j=0; j<length(window[i]); j++//遍历各分片

4)　　if(window[i][j]是 Second 请求)

5)　　count_value[i]++//计算 count_value 值

6)　　end if

7)　end for

8)　count_value$_V$←取 count_value[]的平均值

9)　$\text{AX}\leftarrow\max\left(\frac{\text{count\_value}_i}{\text{count\_value}_V}\right)$　//从所有比值中获得最大值

10) end for

11) return AX//输出 AX 值

当 T 取不同值时，AX 值的范围如表 1 所示。

表 1　　**当 T 取不同值时 AX 值的范围**

T/min	Fast-flux 域名 AX 值范围	88%的非 Fast-flux 域名 AX 值范围
30	2.82~4.36	1.70~6.40
20	4.24~5.67	1.76~5.38
15	3.76~6.08	1.88~4.80
10	5.08~5.61	2.11~6.62
5	6.78~11.22	2.46~7.38

从表 1 可以看出，当 T=5 min 时，AX 值可以更好地表示出一个域名是否为 Fast-flux 域名。这也证实了所发现的重要特性——大部分时间极少被访问，特定时间集中被访问。

数据预处理模块的流程及原理如下。

1) 由 Fast-flux 域名的定义可知，其在一定时间内单域名映射的 IP 地址量很大，故对于待检测的 DNS 流量，将解析 IP 地址数量不足 N_{IP} 的域名筛除。

2) Fast-flux 恶意域名的另一个特点是 IP 地址所归属的国家或地区分布广，所以对于待检测的

DNS 流量，将解析 IP 地址对应的国家或地区数仅为 N_C 的域名筛除。

3)根据所发现的时间分布特性，对于待检测的 DNS 流量，将其中 AX 值低于 N_{AX} 的域名筛除。

为确定上述 N_{IP}、N_C 和 N_{AX} 的取值，对来自微步在线等第三方平台上的 Fast-flux 域名样本进行分析。最终确定了在预处理模块中 3 个参数的取值分别为 N_{IP}=12、N_C=1 和 N_{AX}=6。

3.2 特征提取

根据已有工作进行归纳和改进，并结合 Fast-flucos 的基本思想，拟采用以下 7 组共计 506 个单特征[21-23]。

1) F1 TTL 特征

① 最大 TTL 值。

② 最小 TTL 值。

③ 不同 TTL 值的个数。

2) F2 IP 地址特征

① 解析 IP 地址的个数。

② 解析 IP/65536（IP 地址前两段）的个数。

3) F3 子域名特征

① 子域名的个数。

② 子域名的长度标准差。

③ 共享域名的个数。

4) F4 数字证书字节数

5) F5 地理特征

① 解析 IP 地址对应的国家或地区数量。

② 解析 IP 地址对应的国家或地区涵盖的大洲情况。

③ 距离指数 D_Score。

6) F6 国家向量表

7) F7 时间向量表

其中，共享域名的含义简述如下。若某域名的解析 IP 地址列表中的某些 IP 地址也被其他域名所解析，则这些域名称为该域名的共享域名[24]。

TTL 特征。通过调研[19]发现，正常域名的 TTL 值往往在 1 天以上，而恶意域名为了使自己更加隐蔽，通常会设置很小的 TTL 值。有 60%以上正常域名的 TTL 值是在 1 200 s 以上的，而恶意域名的 TTL 值通常会小于 1 000 s，虽然 Fast-flucos 只关注恶意域名中的 Fast-flux 域名，但是 TTL 特征仍不失为一个可靠的特征。

数字证书字节数。正常域名在注册时通常具有完备的 WHOIS 注册信息，而恶意域名的注册信息通常是很简略或随机的[19]。WHOIS 信息中有一项是数字证书，而恶意域名没有或信息很少，所以数字证书的字节数就可以直接作为一项有力的特征。

针对 Fast-flux 恶意域名的特有特点，又为每个域名定义了距离指数 D_Score、206 维的国家或地区向量表和 288 维的时间向量表这 3 组特征。

3.2.1 距离指数 D_Score

众多已发表的论文中都提到了 Fast-flux 域名的特点是映射的 IP 地址归属的地理位置（国家或地区）分布的范围广并且分散度高，但是均未提及与具体的国家或地区分布频次和量化的地理广度信息相关的特征，所以根据 Fast-flux 恶意域名样本与非 Fast-flux 域名样本各自解析的 IP 地址归属国家或地区（下文简称“解析国家或地区”）的地理广度信息，为每个域名定义了一个特征，称为距离指数 D_Score。其表征了一个域名的解析国家或地区在地理位置分布上的广度与分散程度，为了计算此特征，需要先获取全部有国际顶级域名后缀的国家和地区（这里共计 206 个）的行政中心的经纬度信息，并都采用弧度制表示。利用式(1)球面上两点距离公式，便可以得到全部 206 个国家或地区的两两间距离表。这样，再根据式(2)便可求得某个域名的 D_Score（即欧氏距离）。

$$D_{12}=R\cos^{-1}[\sin y_1\sin y_2+\cos y_1\cos y_2\cos(x_1-x_2)] \tag{1}$$

其中，R 是地球半径，(x_1,y_1)和(x_2,y_2)分别是球面上两点的弧度制坐标。

$$\mathrm{D\_Score}=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{D_{ij}}{n} \tag{2}$$

其中，D_{ij} 表示该域名中的第 i 个解析国家或地区与第 j 个解析国家或地区行政中心坐标间的球面距离，n 表示该域名解析国家或地区的数量。

3.2.2 国家或地区向量表

本文统计了数量比为 5 000∶5 000 的正负样本中每一个域名分别解析到 206 个不同国家或地区的次数，其中正样本选自微步在线等第三方平台上的 Fast-flux 域名样本，负样本是从 DNS 流量中随机选取的 5 000 个不重复的非 Fast-flux 域名。从统计结果可以看出，各不同国家或地区被解析到的频次与发起解析请求的域名是否为 Fast-flux 域名的关联度很高，因此国家或地区向量表这一组特征被加入进来。该特征的具体表示形式为，对于某给定域名，

获取其所有的解析国家或地区，统计全部 206 个国家或地区各自出现的次数，从而形成这个域名的国家或地区向量表，即一个 206 维的向量，其每个值都表示该域名解析到对应国家或地区的次数。

3.2.3　时间向量表

基于已经过验证的 Fast-flux 域名区别于 CDN 等正常域名的重要特性——大部分时间极少被访问，特定时间集中被访问，加入时间向量表特征。若根据 3.1 节中相关概念的定义和结论，将 DNS 流量按照每 5 min 进行分片，则该特征的具体表示形式为，对于某给定域名，获取该域名在其出现次数最多的一天中的 count_value[]，即一个 288 维的向量，其每个值都表示该域名在其出现次数最多的一天中对应的时间分片内的 count_value。

3.3　特征选择和分类器

根据已总结出的 F1~F7 这 7 组特征，分别使用各组特征的不同组合，并都使用 3 种不同的机器学习方法（朴素贝叶斯、支持向量机和逻辑回归），利用 scikit-learn 的机器学习工具库构建机器学习模型，使用引入了 NumPy 包和 Pandas 包的 Python 语言完成实验。在计算国家和地区间的距离时，使用了 Microsoft Excel 并编写了 VBA 语言的宏。对每种特征组合情况都进行了 10 折验证计算，取 10 次的平均值作为最终的检测结果。最后列出其中平均效果最好的 9 种特征组合，如图 4 所示。

从图 4 可以看到，在各种特征组合的情况下，使用逻辑回归算法取得了最佳效果。特征 F6 和 F7 在图 4 中频繁出现，足见其重要作用。然后尝试使用深度神经网络在最佳特征组合条件下进行分类效果测试，先设置隐藏层的层数为 2，根据人为经验对各种神经元数组合进行实验，得到了一组最理想的组合，即（22, 5），此时 10 折验证的输出结果显示召回率是 0.955 2、精确率是 0.933 6、ROC_AUC 是 0.978 956。参考此结果继续调试当隐藏层层数分别为 3~7 时的情况，分别寻找最佳的检测结果，如图 5 所示。

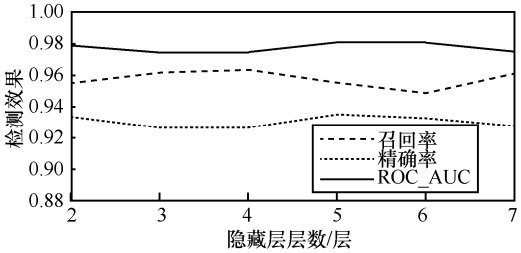

图 5　使用不同隐藏层层数的深度神经网络的分类效果对比

从图 5 可以看出，随着隐藏层层数的增加，检测效果并没有明显变化，当隐藏层层数为 5 时性能最佳，此时召回率是 0.955 4、精确率是 0.935 2、ROC_AUC 是 0.981 0。可见，在 Fast-flux 样本数量不够庞大、所提取的特征已经很具象化的条件下，深度学习并没有取得比常规机器学习更好的检测效果，使用逻辑回归分类器已经达到了较好的性能。

3.4　关联匹配

将从分类器得到的输出结果和原输入 DNS 流量中的域名集合进行“关联匹配”，具体含义如下。

1)“关联”指的是注册者信息关联。在本节中定义从分类器中输出的域名为集合 C，利用 C 中的域名关联 WHOIS 注册者信息来二次挖掘可能被漏检的

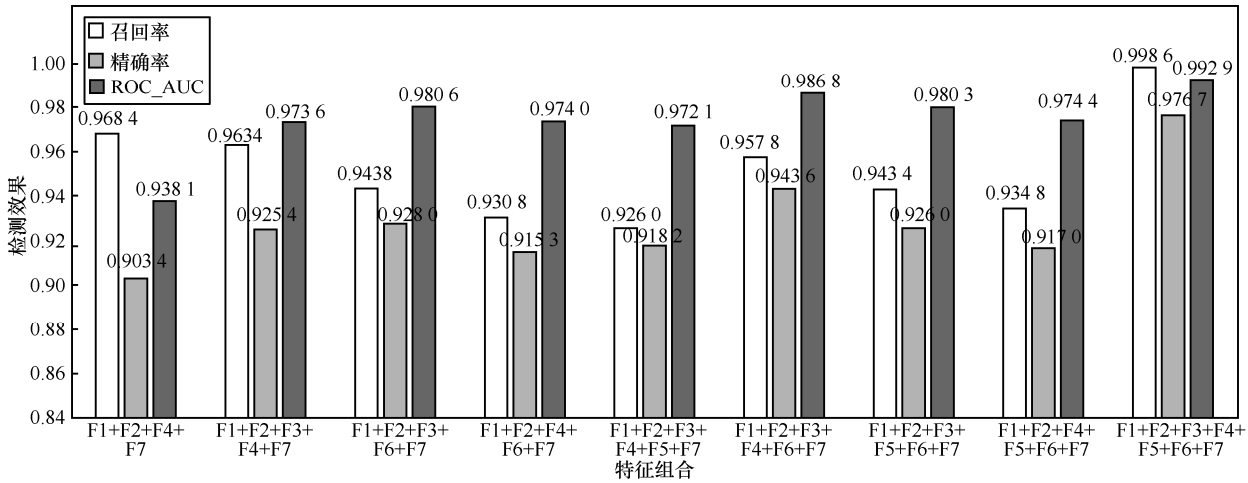

图 4　使用不同的特征组合的分类效果对比

Fast-flux 域名。在原输入 DNS 流量中的域名集合（本节定义为集合 P）中，将与 C 中的域名具有相同注册者的域名都分离出来，在本节定义为集合 S。

2)“匹配”指的是“0-z”化后的字符串匹配。某些个人或集体可能用不同的注册者信息注册了一系列域名用于恶意行为，为便于管理，其中很多域名被命名为仅字母不变而变换数字或仅数字不变而变换字母的形式，如果仅凭注册者信息则无法发现这部分域名。事实上这些域名都普遍指向了相同的 IP 地址集合，即很可能使用了 Double-flux 技术用以减小单点故障的可能性。所以找到具备这种特点的域名，则可以认为其同样是 Fast-flux 域名，在实验中也取得了良好的效果。倘若 C 和 S 中共有域名 m 个，待字符对比的域名有 n 个，则单次对比的次数将是一个时间复杂度为 $O(mn)$的问题，为此，需要引入字符串“0-z”化的概念。

于是，“关联匹配”算法的步骤如下。

步骤 1　将 P 中同时包含字母和数字字符的每个域名中的数字字符都变成 0，成为集合 P_0；除顶级域名外的字母字符都变成 z，成为集合 P_1。对字符串的这种操作称为“0-z”化，如图 6 所示。

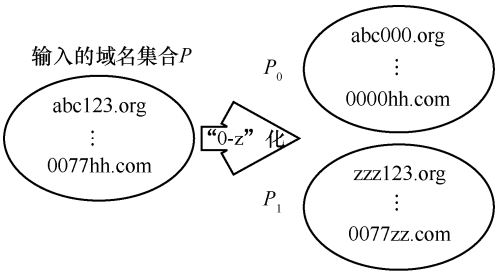

图 6　步骤 1 过程

步骤 2　将 C 和 S 中所有同时带有字母和数字字符的域名连接成一个长字符串 T 并“0-z”化，如图 7 所示。

步骤 3　将 P_0 中的每个字符串分别与 T_0 进行字符串匹配，将 P_1 中的每个字符串分别与 T_1 进行字符串匹配，如图 8 所示。

步骤 4　根据所有匹配成功的字符串，在 P、P_0 和 P_1 中都移除相应的域名/字符串，从 P 中移除的部分独立成为集合 S'。然后将 S 内的全部域名转移进 C 中，如图 9 所示。

步骤 5　在 P 中找到所有与 S' 内注册者相同的域名保存至集合 S，并从 P_0 和 P_1 中移除相应的字符串。然后将 S' 内的全部域名转移到 C 中，如图 10 所示。

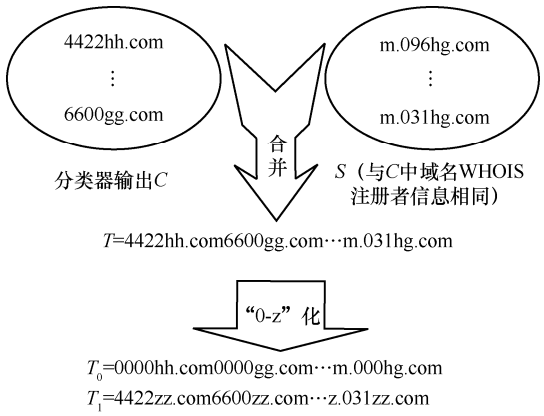

图 7　步骤 2 过程

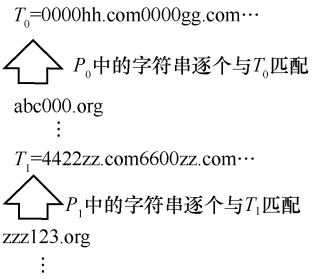

图 8　步骤 3 过程

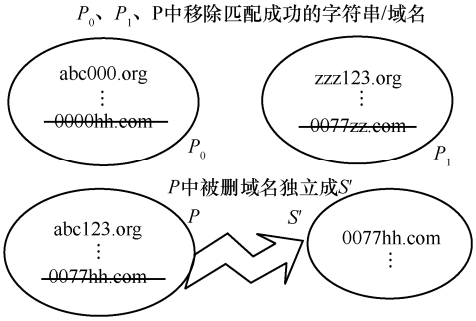

图 9　步骤 4 过程

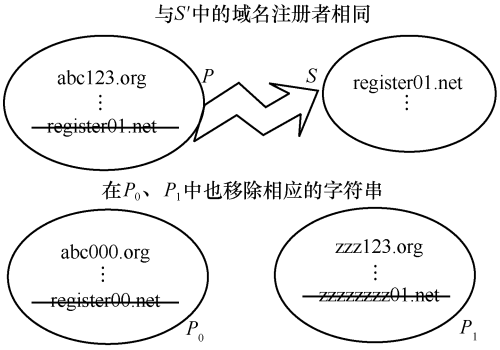

图 10　步骤 5 过程

步骤 6　将 S 中所有同时带有数字和字母字符的域名连接成一个长字符串 T 并“0-z”化，重复步骤 3~步骤 5。

步骤 7　重复步骤 6 直到 C 内域名不再增加。

通过以上处理，将时间复杂度为 $O(mn)$的逐个字符串比较问题转化为时间复杂度为 $O(n)$的字符串匹配问题。而且，因为在“0-z”化后的字符串中存在大量重复的字符，很利于字符串匹配算法的快速运行。整个“关联匹配”算法就是通过 WHOIS 注册者信息和特定的字符串匹配算法相互迭代，使 Fast-flux 域名集合 C 得到不断扩充，从而在分类器分类之后可以再发现一部分漏检的 Fast-flux 域名。

4　性能评估

4.1　分类效果

首先，通过如表 2 和图 11 所示最佳特征组合的逻辑回归分类器详细的 10 折验证结果可以看出，模型的稳定性较好。

表 2　　Fast-flucos 分类器 10 折验证结果

10 折组号	召回率	精确率	ROC_AUC
1	0.996 0	0.981 0	0.992 6
2	1.000 0	0.983 0	0.993 7
3	1.000 0	0.975 0	0.992 3
4	1.000 0	0.977 0	0.994 7
5	0.99 8	0.974 0	0.993 2
6	0.99 8	0.971 0	0.992 5
7	1.000 0	0.971 0	0.993 0
8	0.99 6	0.980 0	0.992 2
9	0.99 8	0.972 0	0.990 1
10	1.000 0	0.983 0	0.994 2
平均	0.998 6	0.976 7	0.992 9

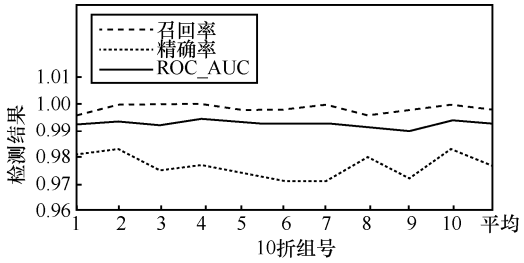

图 11　Fast-flucos 分类器 10 折验证结果

然后，在参考文献中较知名且也同为针对 Fast-flux 域名检测的是文献[13]提出的 GRADE 方法，以此来进行对比实验。GRADE 方法共使用 4 个特征：不同的 A 记录数量 n_A、不同的 ASN 数量 n_{ASN}、先前节点域名熵 e_{DPN}、往返时间标准差 σ_{RTT}，再建立一个计算式：$f(x)=\omega_1 n_A+\omega_2 n_{\mathrm{ASN}}+\omega_3 e_{\mathrm{DPN}}+\omega_4 \sigma_{\mathrm{RTT}}-\beta$。利用真实编码的遗传算法并结合已知标签的样本确定 5 个参数 ω_1、ω_2、ω_3、ω_4 和 β 的值，使当目标域名是 Fast-flux 域名时对应的 $f(x)$ 值均为正数，当目标域名是良性域名时对应的 $f(x)$ 值均为负数。此外，又根据条件分别复现了检测恶意域名最经典方法之一的 EXPOSURE[18]和如何检测 Fast-flux 域名的文献[19]中基于算法生成域名的检测方法（AAGD，approach based on algorithmically generated domain）。使用相同的实验数据，得到如图 12 所示的实验对比结果。

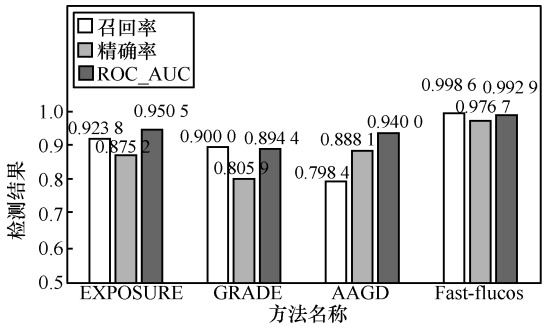

图 12　4 种方法的实验对比结果

GRADE 方法在其原文中的准确率达到了 98%，而在本实验中的精确率却较不理想。这说明该方法在稳定性方面稍弱一些，导致这一结果的可能原因简述如下。

1)采用的特征过少。

2)所使用特征中的 σ_{RTT} 过分依赖于网络环境的稳定性，若网络稍有波动，则 σ_{RTT} 出现的偏差便会影响最终域名判定的结果。

对于 EXPOSURE 的表现，很可能要归咎于它所选用的特征对 Fast-flux 域名的针对性不强。

导致 AAGD 检测效果较不理想的可能原因如下。

1)其 DNS 流量数据来源于教育网环境，不能很好地适应常规网络环境的 DNS 流量。

2)虽然在特征提取方面考虑到了 Fast-flux 域名，但是聚类的过程过于针对 DGA 生成域名。

3)其在特征选择环节并没有在每组特征组合中都分别尝试所有备选机器学习方法，可能错失了更适合的机器学习方法。经实测发现，如果不使用 SVM 算法，实际分类效果确实有了明显的提升。

在 Fast-flucos 中，根据域名解析国家或地区的频次而生成的国家或地区向量表特征起到了很大作用，这可能与各个国家或地区的总体网络普及度、网民数量以及网络犯罪成本的不同有关。而 Fast-flux 域名的解析国家或地区众多，也恰恰使这种在解析国家或地

区分布上的规律可以更加明显地体现出来。

4.2 资源开销

对各方法的内存与时间开销进行对比，考虑到有些方法使用的特征需要在线获取，如 Fast-flucos 的 WHOIS 注册者信息、GRADE 的 σ_{RTT} 等。而第三方网站又对用户访问频率做出了一些限制，为了使算法的执行时间不被这些因素所影响，先将这些特征全部保存到本地，再使用前文的训练数据集来执行每一个算法，得到了如表 3 所示的运行结果。

表 3 不同检测方法的资源开销对比

方法	常驻内存峰值/MB	运行时间/s	时间开销相对值
Fast-flucos	212.6	55.3	1
EXPOSURE	196.7	7.0	0.126
GRADE	194.5	30.7	0.555
AAGD	419.1	364.4	6.596

为了使对比更加清晰，将 Fast-flucos 的运行时间定为单位 1，并以此折算出另外 3 种方法的相对时间开销。从表 3 可以看出，除 AAGD 外，其他 3 种方法的内存开销大抵相同。AAGD 的高常驻内存峰值主要由于 SVM 算法本身的内存占用就相对较高。另外，虽然 EXPOSURE 的检测效果略弱于 Fast-flucos，但是在时间开销上非常出色，而 AAGD 过长的时间开销可能要归咎于聚类的过程及运行速度相对较慢的 SVM 算法。

4.3 实际测试

使用我国某主流通信运营商在上海地区于 2017 年 11 月 29 日的真实 DNS 流量来验证 Fast-flucos 的实际检测效果，共输出 1 215 个域名，经第三方平台查询，并结合已有的 Fast-flux 样本，得到其中 Fast-flux 域名一共有 1 186 个，准确率约为 98%，其中依靠“关联匹配”发现的 Fast-flux 域名有 72 个，被误判的域名约为 29 个，主要由广告服务网站域名等组成。这说明这两类网站的域名在所选取的特征上可能也存在许多与 Fast-flux 恶意域名相似之处，可以考虑作为未来工作的新方向。

5 结束语

因以往的 Fast-flux 域名检测方法在稳定性、针对性和对常规真实 DNS 流量环境的普适性方面存在不足，提出了一种基于真实 DNS 流量的 Fast-flux 恶意域名检测方法——Fast-flucos。该方法由预处理、特征提取、分类器和关联匹配 4 个模块组成。在预处理模块加入了异常过滤步骤；在特征提取模块加入了专门针对 Fast-flux 域名检测的 D_Score、国家和地区向量表和时间向量表特征，共计提取 506 个单特征；使用包括深度学习在内的多种机器学习方法进行实验，确定最佳分类器和特征组合；关联匹配模块可以结合 WHOIS 注册者信息和一种简单的字符串匹配算法借助分类器的输出结果找出更多输入 DNS 流量中被漏检的 Fast-flux 域名。通过以上举措，Fast-flucos 最终较好地弥补了上述不足。通过实验与以往的方法进行检测效果对比，发现 Fast-flucos 在召回率、精确率和 ROC_AUC 均高于另外 3 种方法的前提下，其在内存开销方面的表现与其他方法基本相同，仅在时间开销上稍弱于 EXPOSURE 和 GRADE。最后使用我国某主流通信运营商在上海地区的真实 DNS 流量进行了实际测试，准确率约为 98%，这表明 Fast-flucos 可以普遍适用于常规真实互联网环境下的 DNS 流量。

参考文献：

[1] ZHAUNIAROVICH Y, KHALIL I, YU T, et al. A survey on malicious domains detection through DNS data analysis[J]. ACM Computing Surveys, 2018, 51(4): 67.

[2] ALMOMANI A. Fast-flux hunter: a system for filtering online fast-flux botnet[J]. Neural Computing and Applications, 2018, 29(7): 483-493.

[3] ZHOU C V, LECKIE C, KARUNASEKERA S. Collaborative detection of fast flux phishing domains[J]. Journal of Networks, 2009, 4(1): 75-84.

[4] ZHOU C V, LECKIE C, KARUNASEKERA S, et al. A self-healing, self-protecting collaborative intrusion detection architecture to trace-back Fast-flux phishing domains[C]//IEEE Network Operations and Management Symposium Workshop. Piscataway: IEEE Press, 2008: 321-327.

[5] AL-DUWAIRI B N, AL-HAMMOURI A T. Fast flux watch: a mechanism for online detection of fast flux networks[J]. Journal of Advanced Research, 2014, 1(3): 1-7.

[6] MARTINEZ-BEA S, CASTILLO-PEREZ S, GARCIA-ALFARO J. Real-time malicious fast-flux detection using DNS and bot related features[C]// 2013 Eleventh Annual International Conference on Privacy, Security and Trust. Piscataway: IEEE Press, 2013: 369-372.

[7] CAGLAYAN A, TOOTHAKER M, DRAPEAU D, et al. Real-time detection of fast flux service networks[C]//Proceedings of the Cybersecurity Applications & Technology Conference for Homeland Security. 2009: 285-292.

[8] NAZARIO J, HOLZ T. As the net churns: fast-flux botnet observations[C]//Proceeding of 3rd International Conference on Malicious and Unwanted Software (MALWARE). 2008: 24-31.

[9] CAGLAYAN A, TOOTHAKER M, DRAPAEAU D, et al. Behavioral patterns of fast flux service networks[C]//Proceeding of the 43rd Hawii International Conference on System Sciences (HICSS). Piscataway: IEEE Press, 2010: 1-9.

[10] HU X, KNYSZ M, SHIN K G. Measurement and analysis of global IP-usage patterns of fast-flux botnets[C]//Proceeding of IEEE INFORCOM. Piscataway: IEEE Press, 2011: 15.

[11] PASSERINI E, PALEARI R, MARTIGNONI L, et al. FluXOR: detecting and monitoring Fast-flux service networks[C]//Proceeding of the 5th Conference on Detection of Intrusion and Malware & Vulnerability Assessment(DIMVA). Berlin: Springer, 2008: 186-206.

[12] PERDISCI R, CORONA I, DAGON D, et al. Detecting malicious Flux service networks through passive analysis of recursive DNS traces[C]//Twenty-Fifth Annual Computer Security Applications Conference. Los Alamitos: IEEE Computer Society, 2009: 311-320.

[13] LIN H T, LIN Y Y, CHIANG J W. Genetic-based real-time fast-flux service networks detection[J]. Computer Networks, 2013(57): 501-513.

[14] HUANG S Y, MAO C H, LEE H M. Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection[C]//Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. New York: ACM Press, 2010: 101-111.

[15] HOLZ T, GORECKI C, RIECK K, et al. Measuring and detecting Fast-flux service networks[C]//In Symposium on Network and Distributed System Security. 2008: 1-12.

[16] KNYSZ M, HU X, SHIN K G. Good guys vs. bot guise: mimicry attacks against fast-flux detection systems[C]//Proceeding of IEEE INFORCOM. Piscataway: IEEE Press, 2011: 1844-1852.

[17] HSU F H, WANG C S, HSU C H, et al. Detect Fast-flux domains through response time differences[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(10): 1947-1956.

[18] BILGE L, KIRDA E, KRUEGEL C, et al. EXPOSURE: finding malicious domains using passive DNS analysis[C]//Proceedings of the Network and Distributed System Security Symposium. 2011: 1-17.

[19] 臧小东, 龚俭, 胡晓艳. 基于 AGD 的恶意域名检测[J]. 通信学报, 2018, 39(7): 15-25.
ZANG X D, GONG J, HU X Y. Detecting malicious domains based on AGD[J]. Journal on Communications, 2018, 39(7): 15-25.

[20] FAKERI-TABRIZI A, NGUYEN T, LIU H L, el al. Analyzing DNS requests for anomaly detection: US 20160065611A1 [P]. (2016-03-03) [2019-10-31].

[21] LEI K, FU Q, NI J, et al. Detecting malicious domains with behavioral modeling and graph embedding[C]//2019 IEEE 39th International Conference on Distributed Computing Systems. Piscataway: IEEE Press, 2019: 601-611.

[22] SUN X, TONG M, YANG J, et al. HinDom: a robust malicious domain detection system based on heterogeneous information network with transductive classification[C]//22nd International Symposium on Research in Attacks, Intrusions and Defenses. Berkeley: USENIX Association, 2019: 399-412.

[23] SHI Y, CHEN G, LI J. Malicious domain name detection based on extreme machine learning[J]. Neural Processing Letters, 2018, 48(3): 1347-1357.

[24] 周昌令, 陈恺, 公绪晓, 等. 基于 Passive DNS 的速变域名检测[J]. 北京大学学报(自然科学版), 2016, 52(3): 396-402.
ZHOU C L, CHEN K, GONG X X, et al. Detection of Fast-flux domains based on passive DNS analysis[J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2016, 52(3): 396-402.

[作者简介]

韩春雨（1990– ），男，黑龙江鹤岗人，南开大学博士生，主要研究方向为网络与信息安全。

张永铮（1978– ），男，黑龙江哈尔滨人，博士，中国科学院信息工程研究所研究员、博士生导师，主要研究方向为网络安全态势感知。

张玉（1981– ），男，浙江湖州人，南开大学副教授、硕士生导师，主要研究方向为网络安全、数据安全、数据挖掘等。