

中图分类号：TP393.08

论文编号：203130304

学科分类号：520.1060

密 级：

天津理工大学研究生学位论文

基于深度学习的 DGA 域名检测方法 研究

（申请硕士学位）

一级学科：网络空间安全

研究方向：DGA 恶意域名检测

作者姓名：王天宇

指导教师：王春东 教授

2023 年 2 月

**Thesis Submitted to Tianjin University of Technology
for the Master's Degree**

**Research on DGA Domain Names
Detection Method Based on Deep
Learning**

By
Tianyu Wang

Supervisor
Chundong Wang

February 2023

摘要

近年来,网络空间中的恶意攻击行为的数量呈高速上升趋势,高频率且多样化的恶意攻击对广大互联网用户的个人数据、财产安全产生了严重的威胁。僵尸网络是目前影响力和破坏力极大的一种网络恶意攻击平台,其通常使用 DNS 服务与命令和控制(C&C)服务器进行通信。为了绕过黑名单机制的防御,攻击者通常使用域名生成算法(DGA)生成用于通信连接的新域名。

目前,主流的 DGA 域名检测技术采用可自动提取特征的深度学习方法。然而,基于深度学习的检测方法仍存在两个亟需解决的问题。一是 DGA 域名具有短文本的特点,现有模型对其文本信息的利用率低,导致多分类效果不佳。二是新型 DGA 家族层出不穷,它们的生成方式多样化,且字符随机性低,与良性域名在字符分布和组成上十分相似,现有检测方法对其检测效果不佳。本文针对以上两方面的问题展开研究,主要的贡献和创新点如下:

(1) 针对目前部分检测方法对域名信息利用率低,导致多分类效果不佳的问题,提出了一种基于深度学习的 DGA 域名分类模型(PCBGA-DGA)。该模型利用并行的卷积神经网络(PCNN)和结合注意力机制的双向门控循环单元(BiGRU-Att),分别提取域名序列的局部特征,和包含字符权重信息的时序依赖特征。实验结果表明,相比传统的深度学习模型,该模型对 DGA 域名的多分类效果更好,在分类两种基于单词列表生成的 DGA 域名时也有明显的优势。

(2) 针对现有检测方法检测新型 DGA 域名的效果不佳的问题,提出了一种新型 DGA 域名(CLR-DGA)的生成方法及其防御措施。CLR-DGA 通过基于良性域名的字符级替换,生成难以检测的 DGA 域名。之后本文基于 CLR-DGA 展开了对抗攻击和防御实验。在对抗攻击实验中,比较 CLR-DGA 与 3 种已知的 DGA 和 2 种新型 DGA,观察 5 种深度学习分类器对以上 6 种 DGA 的检测效果。实验结果表明,有 4 种分类器检测 CLR-DGA 的效果最差。在对抗防御实验中,使用额外的 1 万个 CLR-DGA 域名和 1 万个良性域名扩充对抗攻击实验中使用的训练数据集,对 5 种深度学习分类器进行了再训练。实验结果表明,对抗训练方法可以在一定程度上增强深度学习分类器抵御来自 CLR-DGA 的攻击,进而提升分类器的鲁棒性。

关键词: 僵尸网络, 域名生成算法, 域名分类, 神经网络, 对抗攻击

Abstract

In recent years, the number of malicious attacks in cyberspace is on the rise at a high speed. High frequency and diversified malicious attacks pose a serious threat to the personal data and property security of Internet users. Botnet is a network malicious attack platform with great influence and destructive power at present. It usually uses DNS services to communicate with command and control (C&C) servers. To avoid the defense of blacklist mechanism, attackers usually use the domain generation algorithm (DGA) to generate new domain names for communication connections.

Nowadays, the mainstream DGA domains detection technology adopts deep learning methods that can automatically extract features. However, the detection methods based on deep learning still have two problems that need to be solved urgently. First, DGA domain names have the characteristics of short text, and the existing models have low utilization of their text information, resulting in poor multi-classification effect. Second, new DGA families emerge in endlessly. Their generation methods are diversified, and the character randomness is low. They are very similar to benign domain names in character distribution and composition, and the existing detection methods have poor detection effects. This paper focuses on the above two aspects, and the main contributions and innovations are as follows:

(1) Some of the current detection methods have low utilization of domain name information, resulting in poor multi-classification effect. Aiming at this problem, a DGA domain name classification model based on deep learning (PCBGA-DGA) is proposed. In this model, parallel convolutional neural network (PCNN) is used to extract local features of domain name sequence, and bidirectional gated recurrent unit combined with attention mechanism (BiGRU-Att) is used to extract time-series-dependent features of domain name sequence containing character weight information. The experimental results show that, compared with the traditional deep learning model, this model has a better effect on the multi-classification of DGA domain names, and has obvious advantages in classifying two DGA domain names based on word lists.

(2) The existing detection methods are not effective in detecting new DGA domain names. For this problem, this paper proposes a new generation method of DGA domain names (CLR-DGA) and its defense measures. CLR-DGA generates DGA domain names, which are difficult to be detected, by character-level replacement based on

benign domain names. Then, based on the CLR-DGA, this paper conducts the research on adversarial attack and defense. In the adversarial attack experiment, CLR-DGA is compared with three known DGAs and two new DGAs, to observe the detection effect of five depth learning classifiers on the above six DGAs. The experimental results show that there are four kinds of classifiers that have the worst performance in detecting CLR-DGA. In the adversarial defense experiment, the training data set used in the adversarial attack experiment was expanded with 10000 additional CLR-DGA domain names and 10000 benign domain names, and five deep learning classifiers were retrained. The experimental results show that the adversarial training method can enhance the depth learning classifier to resist attacks from CLR-DGA to a certain extent, thereby improving the robustness of the classifier.

Key words: Botnet, Domain generation algorithm, Domain name classification, Neural network, Adversarial attack

目 录

第一章 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	2
1.2.1 基于传统机器学习的检测方法.....	2
1.2.2 基于深度学习的方法.....	5
1.2.3 基于对抗攻击的新型 DGA 生成和防御方法.....	6
1.3 论文主要工作.....	7
1.4 论文组织架构.....	8
第二章 相关理论及技术原理	10
2.1 僵尸网络.....	10
2.2 DGA 恶意域名	11
2.2.1 域名系统.....	11
2.2.2 域名生成算法.....	12
2.3 深度学习.....	13
2.3.1 卷积神经网络.....	13
2.3.2 循环神经网络.....	14
2.3.3 注意力机制.....	16
2.4 本章小结.....	17
第三章 基于深度学习的 DGA 域名分类模型.....	18
3.1 PCBGA-DGA 模型架构	18
3.1.1 预处理.....	19
3.1.2 嵌入层.....	19
3.1.3 特征提取层.....	20
3.1.4 输出层.....	22
3.2 实验与分析.....	22
3.2.1 实验环境.....	22
3.2.2 数据集和评估指标.....	23
3.2.3 多分类对比实验.....	25
3.2.4 多分类消融实验.....	28
3.2.5 DGA 家族关联分析	30
3.3 本章小结.....	32
第四章 基于字符级替换的新型 DGA 域名生成及防御方法	33
4.1 对抗攻击.....	33
4.2 CLR-DGA.....	34

4.3 实验与分析.....	38
4.3.1 实验环境与分类模型.....	38
4.3.2 数据集.....	38
4.3.3 评估指标.....	40
4.3.4 CLR-DGA 有效性验证	40
4.3.5 CLR-DGA 防御	42
4.4 本章小结.....	44
第五章 总结与展望	45
5.1 本文总结.....	45
5.2 未来展望.....	45
参考文献.....	47

第一章 绪论

1.1 研究背景和意义

互联网极大地便利了信息交换,促使人类社会正式步入了信息化时代,是现代科技迅猛进步、经济高速发展的重要保障。根据 2022 年 8 月发布的《第 50 次中国互联网络发展状况统计报告》^[1],从 1997 年 10 月到 2022 年 6 月,我国网民数量由 62 万增长至 10.51 亿,占全球互联网用户的五分之一以上,规模居全球之首。我国十亿以上的网民,构成了全球最为庞大且充满发展潜力的数字社会。

然而,互联网在为人类提供多样化的便利服务的同时,其安全形势并不乐观。计算机与互联网在设计之初,以实现信息传输等核心功能为主,并对使用者抱有充分的信任,因而对安全因素的考虑不足,致使网络空间存在大量的漏洞。由《2022 上半年网络安全漏洞态势观察》^[2]可知,2022 年上半年新增通用型漏洞信息共计 12466 条,有 53% 的漏洞是高危甚至超危的漏洞。这些漏洞的存在,使得攻击者能够在未授权的情况下顺利地执行网络犯罪^[3]行为,进而牟取私利。网络犯罪包含多种恶意活动^[4],如发送垃圾邮件,传播和部署恶意软件,窃取敏感信息,分布式拒绝服务(Distributed denial of service, DDoS),高级持续性威胁(Advanced persistent threat, APT)^[5]攻击等。这些恶意活动大多基于僵尸网络(Botnet)实施。

僵尸网络是当今威胁互联网安全的重大威胁之一。《2021 年上半年中国互联网络网络安全监测数据分析报告》^[6]指出,在我国境内存在 2307 个规模超过 100 台主机的僵尸网络,其中 68 个僵尸网络的规模多于 10 万台。僵尸网络的恶意攻击行为,对我国网络空间的安全发起了巨大的挑战。

僵尸网络由三部分组成:管理僵尸网络的黑客(Botmaster),命令和控制(Command and control, C&C)服务器,僵尸主机(Bots)。Botmaster 负责使用恶意软件感染并获得僵尸主机,对它们远程发出攻击命令。僵尸主机通过恶意软件,接收命令并执行恶意活动。C&C 服务器负责作为 Botmaster 和僵尸主机之间的 C&C 通信中介, Botmaster 通过 C&C 服务器向 Bots 发送指令,进而发动各种恶意攻击。其中, C&C 通信是 Botmaster 控制 Bots 进行各种攻击行为的关键信息交互方式,因此 Bots 与 C&C 服务器通信的稳定性决定了僵尸网络的鲁棒性^[7]。

大多数恶意软件需要一种方法在 Bots 与 C&C 服务器之间建立稳定的 C&C 通信信道。早期的恶意软件开发者,习惯于在恶意软件代码中硬编码 C&C 服务

器使用的 IP 地址或域名。然而，安全研究人员可以分析恶意软件产生的网络流量，通过黑名单技术拦截 IP 地址或域名；或通过逆向工程技术，分析恶意软件代码，成功识别其中包含的硬编码 IP 地址或域名信息。为了更好地逃避系统对恶意域名的封锁和拦截，僵尸网络借助域名生成算法（Domain generation algorithm, DGA）来生成恶意域名，即“恶意的算法生成的域名”（Malicious algorithmically generated domain, MAGD）。

通过采用 DGA，Bots 上运行的恶意软件不必依赖固定的域名或 IP 地址列表，而是执行算法在短时间内生成大量可能的 MAGD（每天多达数十万个），并尝试定期查询这些 MAGD。然后，恶意软件控制者即 Botmaster 只需要注册这些 MAGD 中的至少一个，就可以在 Bots 和 C&C 服务器之间建立通信；其它未注册的 MAGD 可以产生噪声流量，增大 C&C 通信的检测难度。为加强 C&C 通信的隐蔽性，C&C 服务器使用的 MAGD 存活时间短，多数只有 1 到 7 天^[8]。并且，攻击者还会采用 Fast-flux 技术^[9]，不断改变 C&C 服务器的 IP 地址与 MAGD 的对应关系，以增加防御人员定位 C&C 服务器的难度。这些因素使得黑名单过滤及逆向工程分析恶意软件的防御方式在效果上十分有限。综上所述，对 MAGD 的检测识别是维护网络空间安全的迫切需要。

1.2 国内外研究现状

DGA 具有高隐蔽性、低成本的特点，因此被广泛地应用于僵尸网络中^[10]。随着僵尸网络的发展以及 DGA 的不断“升级”，经典的黑名单防御手段已经无法识别这些伪随机性较强、存活时间短、数量庞大的 MAGD。文献[11]根据已知的僵尸程序逆向分析 DGA 原理，通过提前抢注的方式阻断 Bots 与 C&C 服务器的通信。逆向工程虽然可以分析 DGA 的原理和其域名生成机制，但作为防御 DGA 的措施仍有不少缺点。首先，逆向工程的人力成本和时间成本高，难以应对快速生成大量域名的 DGA。其次，对同一家族的 DGA，攻击者只需修改少数代码，即可产生 MAGD 变体，使得原先逆向得到的 MAGD 列表几乎失效。之后的研究关注良性域名与 DGA 域名的文本特征差异性，进而识别 MAGD。基于域名文本特征的 DGA 域名检测技术共经历了三个发展阶段：基于传统机器学习的检测方法、基于深度学习的检测方法和基于对抗攻击的新型 DGA 生成及防御方法。

1.2.1 基于传统机器学习的检测方法

继黑名单与逆向工程之后，研究人员采用基于传统机器学习的检测方法，以应对 MAGD 的威胁。基于传统机器学习的检测方法，包括数据集预处理，提取

域名数据特征,训练检测模型和验证模型效果四个步骤。其中,域名数据特征的提取是影响机器学习检测方法效果的关键。提取的域名特征主要包括两类:一是域名数据的内部特征,包括域名长度、TLD 是否合法等域名结构特征,元音字母比率、数字比率等域名语法特征,N-gram 频率分布、相对熵等域名统计特征,生存时间(Time to live, TTL)等 DNS 流量特征;二是域名数据的关联特征,通过图对 DNS 流量数据进行建模,使用节点和边表示域名、IP、主机等网络节点之间的联系。

(1) 基于内部特征的检测方法

基于内部特征的方法直接处理并提取 DNS 流量、域名字符串等数据内部包含的特征,用于模型的训练和检测。该类方法采用分类、聚类或分类与聚类相结合的技术。

基于分类的方法从数据中人工提取特征后,再使用决策树、随机森林等算法对域名进行分类,通常以二分类为主。Bilge 等人^[12]构建了 EXPOSURE 域名检测模型,它从被动 DNS 流量中提取了 15 个特征,包含时间、DNS 响应、TTL 取值、域名字符串本身共四类,并采用 J48 决策树算法对域名进行二分类。但 Peng 等人^[13]指出,EXPOSURE 系统需要 17.45 天的时间延迟,才能积累 20 次某个域名的 DNS 查询流量信息,进而从中提取到对该域名进行分类所需的特征。Schüppen 等人^[14]提出了 FANCI 系统,它利用 DGA 实际注册的域名数量远少于其生成的 MAGD 数量的性质,仅监测被动 DNS 流量中的 NXD 部分,并仅从 NXD 流量中提取共 3 类 21 个特征,使用随机森林算法进行二分类,将 NXD 分类为 MAGD 和 BNXD (Benign NXD)。在亚琛工业大学校园网的一个月的实际应用中,FANCI 发现了 10 组与 DGA 相关的 MAGD,其中至少有 4 组来自全新的 DGA。

基于聚类的方法,基于特定的规则和算法将域名分割为多个簇,每个簇中的域名有一定程度的相似性,这有助于探索域名潜在的类别。Thomas 等人^[15]分析顶级域名运营商 Verisign 授权的 DNS 流量数据,计算 NXD 之间的相似度,然后采用层次聚类对属于同一 DGA 家族或其变体的域名进行分组。Wang 等人^[16]根据 DNS 流量日志中的查询行为,对使用同一 DGA 算法的终端主机,使用 Chinese whispers 算法进行聚类,对 Kraken、Conficker、Cycbot、Murofet 四个 DGA 家族的检测准确率(Accuracy)均达到 0.996 及以上。

由于聚类本质上是一种非监督分类方法,还有一些研究工作将分类和聚类技术结合使用,以实现域名多分类的检测效果,这时良性域名统一为 Non-DGA 类,而 MAGD 继续细分为不同的 DGA 家族。Antonakakis 等人^[17]提出了 PLEIADES 检测系统,它首先使用 X-means 聚类算法,根据 NXD 流量中域名组

成的相似性和查询这些域名的主机，对域名进行聚类；之后通过交替决策树（Alternating decision trees, ADT）方法，将生成的聚类分类为 ADT 训练期间已知的 DGA，若某个聚类无法分类为已知的 DGA，则将其分类为新的第 x 种 DGA，命名为 New-DGA- vx 。PLEIADES 在北美大型互联网服务提供商（Internet service provider, ISP）部署的 15 个月内，发现了 12 种 DGA，其中一半是已知 DGA 的变种，另一半是原先未知的 DGA 家族。Chin 等人^[18]基于 6 个域名词汇特征和 27 个 DNS 流量特征，首先使用 J48 决策树对域名进行二分类；之后通过具有噪声的基于密度的聚类（Density-based spatial clustering of applications with noise, DBSCAN）算法，将恶意域名聚类为不同的家族。

基于内部特征的方法存在一些缺陷：部分特征时效性较差，导致其效果不稳定；人工提取的特征容易被攻击者得知并有针对性地绕过；特征选取和组合的好坏对检测结果有较大影响^[19]。为规避这些缺点，一些研究者对基于关联特征的方法开展了研究。

（2）基于关联特征的方法

基于关联特征的方法将 DNS 流量等信息建模为图，通过图表示域名、IP、主机等网络核心节点之间隐含的关联特征，之后通过机器学习方法检测域名。与内部特征相比，关联特征倾向于描述数据之间隐含的某种稳定的联系，如域名和 IP 地址的对应关系。关联特征对传统特征工程的需求更少，而对使用图对域名数据进行建模和分析的要求更高。

Rahbarinia 等人^[20]提出的 SEGUGIO 系统专注于检测 C&C 域名，它监控两个大型 ISP 网络中的被动 DNS 流量，构建主机和域名的异构关系图，以表示网络中用户的查询行为等关联特征；之后依据这些特征，通过随机森林、逻辑回归等方法分类域名。但 Peng 等人^[13]指出，SEGUGIO 系统需要 16.43 小时的时间延迟，才能收集到足以检测一个 MAGD 的 DNS 流量。Sun 等人^[21]首次将图卷积网络（Graph convolutional network, GCN）应用于网络安全分析，并提出了 HGDOM，一种基于异构图卷积网络方法的恶意域名检测系统。HGDOM 引入异构信息网络（Heterogeneous information network, HIN）对 DNS 响应流量建模为有向无环的异构图，相比域名本身的同构图，异构图丢失的信息更少；图中包含主机（客户端）、IP 地址和域名三种节点，以捕获更丰富的信息；还提出了一种称为 MAGCN 的表示方法，借助基于元路径的注意力机制，同时处理 HIN 中的节点特征和图结构。Li 等人^[22]提出了 HANDOM 系统，在使用 HIN 网络对 DNS 流量进行图建模的基础上，使用分层注意力机制学习不同邻居节点的元路径的重要性以及不同元路径对当前域名节点的重要性。对比实验中，HANDOM 的 F1 分数为 0.9927，高于 HGDOM 等其它对比模型至少 1 个百分点。

相对于黑名单,传统机器学习模型便于维护,也不需要占用过量的数据资源以存储域名黑名单数据;相对于逆向工程,传统机器学习的特征工程步骤性价比更高,可以提取可解释性较好且长期有效的特征,不需反复对同一种 DGA 及其变种进行逆向分析。然而基于传统机器学习的检测方法过于依赖人工特征工程,时间成本及人力成本较高;且部分特征需要等待 MAGD 产生一定的 DNS 流量才能顺利提取,并使得检测率保持较高的水平,这难以满足目前对于 MAGD 实时检测的安全需求。

1.2.2 基于深度学习的方法

深度学习是传统机器学习的改进和提升,在自然语言处理、图像处理、语音识别等许多领域取得了显著成果^[10]。Woodbridge 等人^[23]是最早将深度学习方法引入 MAGD 检测的,他们设计了 Endgame,一种面向 MAGD 的构造简单的长短时记忆(Long short term memory, LSTM)模型。与先前的传统机器学习方法相比,该方法有诸多优点:可以对域名进行实时多分类检测(分类单个域名平均只需 20 毫秒),只需域名字符串数据集而无需其它信息,无需手工提取特征等。Endgame 使研究人员发现了深度学习技术在检测 MAGD 方面的巨大潜力,为 MAGD 检测领域提供了新的研究方向。

受到 Woodbridge 等人的启发,一些工作基于循环神经网络(Recurrent neural network, RNN)结构展开研究,通过 RNN 结构学习域名字符串序列隐含的单向或双向序列信息。Tran 等人^[24]注意到多种类别的 DGA 和良性域名之间的数据不平衡问题,并在 Endgame 方法的基础上,提出了改进的 LSTM.MI 算法。在该模型中,LSTM 网络在其反向传播学习机制中包含成本项,以缓解多分类不平衡问题,使得模型对 DGA 的宏平均精确度(Precision)和召回率(Recall)平均提升了 7%。LSTM.MI 模型分类基于单词的 Suppobox 家族的精确度仅有 0.3542,召回率仅有 0.0934;但与对 Suppobox 的精确度和召回率均为 0 的 Endgame 相比已有了质的提升。Tuan 等人^[25]提出了两种深度学习模型 LA_Bin07 和 LA_Mul07,前者用于二分类,后者用于多分类。他们采用多个包含数百万到数千万域名字符串的数据集对两个模型进行训练和测试,发现两个模型与先前的工作相比均有更高的精度。并且,LA_Bin07 的训练时间最长是 6883 秒,测试时间最长是 204 秒;LA_Mul07 的训练时间最长是 2324 秒,测试时间最长是 55 秒。因此,LA_Bin07 和 LA_Mul07 的训练和测试速度都很快,可适应实时检测的需求。

一些工作尝试基于卷积神经网络(Convolutional neural networks, CNN)结构进行 MAGD 的检测。根据卷积核大小的不同,CNN 结构可学习域名字符串等一维序列数据的局部序列信息,类似自然语言处理领域中的 N-gram 特征。Yu 等人

[26]比较了一系列字符级文本分类算法对 MAGD 的检测效果,发现文献[27]中最初提出的称为 *Invincea* 的并行 CNN 体系结构的检测精度最高,可达 0.9895 的准确率和 0.9801 的召回率,检测单个域名平均需时 0.35 毫秒。刘小洋等人[28]提出了一种基于字符级滑动窗口的深度残差网络,将轻量级深度可分离式卷积应用于 MAGD 检测,相比标准卷积减少了约 56% 的参数,增强了模型检测效率。Zhao 等人[29]提出了一种称为 D3-SACNN 的 MAGD 检测方法,将卷积核为 3、5、7、9 的四层 CNN 与多头自注意力机制相结合,对域名多分类达到了总体 0.9863 的 F1 分数,比其它对比模型高出至少 0.5 个百分点;检测 Suppobox 家族的 F1 分数达到了 0.9510,比其它对比模型高出至少 1.49 个百分点。

还有一些工作尝试集成 RNN 与 CNN,充分学习域名字符串的局部及全局序列信息,以提升检测结果。杜鹏等人[30]为更加充分地利用域名字符串的信息,提出了一种混合词向量,该向量结合了域名的单个字符与双字符组合这两类信息;并将混合词向量、CNN、LSTM 三者相结合,提升了 MAGD 多分类任务的检测精度。Namgung 等人[31]提出了一种基于双向长短时记忆 (Bidirectional LSTM, BiLSTM) 网络的实时的 MAGD 检测方法,并进一步提出了一种集成模型方法,该方法将侧重于局部序列信息的 CNN,和侧重于全局序列信息的 BiLSTM 并联起来,并结合使用注意力机制,从多个角度学习域名字符串隐含的特征信息,并达到了 0.9666 的多分类 F1 分数。Liu 等人[32]提出了一种基于 K-means 算法的序列胶囊网络 LSTM-CAPSNET,该模型使用 BiLSTM 初步提取特征信息并作为胶囊网络的输入数据;并使用 K-means 算法对胶囊网络中的向量特征进行聚类以实现胶囊层中的路由功能。LSTM-CAPSNET 对域名的多分类宏平均 F1 分数比 Endgame 等对比模型高出至少 0.5 个百分点。

与基于传统机器学习的检测方法相比,基于深度学习的检测方法无需人工提取特征,训练后的模型可在平均数毫秒内检测单个域名并得出分类结果[26],一定程度上提升了检测准确率,且黑盒性质提高了攻击者的分析成本。基于深度学习的检测方法已成为 MAGD 检测领域的主流技术,但依旧存在一些问题。其一,多分类时,一些深度学习方法对域名序列信息的利用率低,在分类部分 DGA 域名如 Suppobox 和 Matsnu 时,由于域名字符串随机性低等因素,分类效果不佳,即深度学习模型对域名的多分类效果仍有上升空间。其二,基于深度学习的方法容易受到对抗攻击,在鲁棒性、安全性等方面仍有不足。

1.2.3 基于对抗攻击的新型 DGA 生成和防御方法

对抗攻击针对机器学习模型的脆弱性,使模型错误分类或得出攻击者期望的检测结果。近年来诸多研究表明,在图像识别等领域,深度学习模型容易受到对

抗样本的攻击。而在 MAGD 检测领域,一些研究者生成了可用于攻击机器学习模型的新型 DGA,通过学习已有良性域名和 MAGD 的特征,以模拟真实世界中未知 DGA 难以被检测的特性。新型 DGA 的出现,对深度学习模型的检测质量提出了更高的要求。

与图像识别领域类似,新型 DGA 的生成方法以生成对抗网络(Generative adversarial networks, GAN)为主。Anderson 等人^[33]最早使用 GAN 生成名为 DeepDGA 的新型 DGA,之后用随机森林 MAGD 分类器来对比验证 DeepDGA 与常见的 10 种恶意软件 DGA 的检测效果。结果,常见 DGA 检测成功率至少有 85%,而 DeepDGA 检测成功率仅有 48%,这证明了 DeepDGA 回避机器学习分类器检测的有效性。Yun 等人^[34]提出了一种基于神经语言模型和 WGAN(Wasserstein GAN)的新型 DGA——Khaos。这一研究的关键切入点是,真实的良性域名由可读的音节和首字母缩略词组成,因此可以使用神经语言模型来安排音节和首字母缩略词,以模仿良性域名。此外,Yun 等人发现,将 Khaos 生成的域名对现有的检测方法进行对抗训练,可以提高其检测能力。Zheng 等人^[35]提出了 ShadowDGA,该方法将 GAN 网络与残差网络相结合,其生成器和判别器结合了 1 个一维卷积层和 5 个顺序连接的残差块,在避免大量参数计算的同时,通过残差块可以有效地降低神经网络退化的风险,进而更好地模拟 Alexa 良性域名的分布。与同样基于 GAN 生成的 DeepDGA 相比,FANCI、Endgame 等分类器检测 ShadowDGA 的 F1 分数至少低了 10 个百分点。

一些研究者从其它角度考虑生成新型 DGA 的方法。Spooren 等人^[36]提出了 Deception_DGA,一种利用特征工程的知识分析良性域名而构造的新型 DGA,该新型 DGA 生成的 MAGD 可使随机森林分类器 FANCI 的检测准确率仅为 59.9%,但 LSTM 分类器的检测准确率仍有 85.5%。Peck 等人^[37]提出了 Charbot,这种 DGA 只需随机替换现有良性域名中的两个字符即可生成,并且使得 FANCI 和 LSTM.MI 方法难以对其准确分类。Sidi 等人^[38]提出的 MaskDGA 以 MAGD 作为原始数据,通过基于雅可比矩阵的显着性图攻击(Jacobian-based saliency map attacks, JSMA)方法生成新型 DGA。

目前 MAGD 检测方法以提取域名数据特征的机器学习技术为主,但机器学习技术难以检测以对抗攻击为目标的各类新型 DGA。如何针对新型 DGA 生成的 MAGD 增强相关检测模型的鲁棒性,也将成为未来的主要研究方向之一。

1.3 论文主要工作

随着网络攻击技术的不断迭代发展,僵尸网络的相关技术更加先进且多样化,

且性能更加强大。僵尸网络经常使用 DGA，以逃避防御系统的检测。其中，检测方法对域名字符串信息的利用率低导致的多分类效果不佳的问题，及生成方式多样化且更加难以检测的新型 DGA 的出现，也对现有的域名检测技术提出了挑战。针对以上两大难题，本文开展的具体工作如下：

（1）分析了本文涉及的相关技术理论，如僵尸网络的特性与运行机制、域名字符串的构成、DGA 的分类、适用于域名文本序列的深度学习检测技术等。

（2）根据攻击对象的不同，攻击者通常会采用不同的 DGA。为了帮助网络管理员快速且精准地阻断攻击行为，提高检测模型对 DGA 域名的分类性能尤为重要。本文提出了一种基于深度学习的 DGA 域名分类模型 PCBGA-DGA。该模型利用并行卷积神经网络（Parallel CNN, PCNN）和结合注意力机制（Attention mechanism）的双向门控循环单元（Bidirectional GRU, BiGRU），分别提取域名序列的局部特征和包含字符权重信息的上下文时序依赖特征；之后融合这两种特征信息，使用深度学习模型对公开的域名数据集执行多分类。实验结果表明，该模型在 DGA 域名多分类任务中，加权平均的 F1 分数达到了 0.9343；与四种深度学习对比模型相比，该模型多分类性能综合表现更优，分类 20 种 DGA 域名家族时，有 16 种家族取得了最高的 F1 分数。

（3）生成方式多样化且相比已知 DGA 更加难以检测的新型 DGA 不断产生。探索并预测新型 DGA 家族可能的生成方法，以及提高深度学习技术检测这些新型 DGA 家族的性能十分重要。本文提出了 CLR-DGA，该新型 DGA 通过基于良性域名的字符级替换生成难以检测的域名。之后本文基于 CLR-DGA 展开了对抗攻击和防御实验。在对抗攻击实验中，比较 CLR-DGA 与 3 种已知的 DGA（基于字符的 Bamital 和 Ramnit，基于单词的 Suppobox），和两种新型 DGA（基于深度学习方法的 DeepDGA 和基于字符替换的 Charbot），观察 5 种深度学习分类器对以上 6 种 DGA 的检测效果。在对抗防御实验中，使用额外的 1 万个 CLR-DGA 域名和 1 万个 Tranco 良性域名扩充对抗攻击实验中使用的训练数据集，对 5 种深度学习分类器进行了对抗训练。实验验证了 CLR-DGA 生成的 DGA 域名的有效性，以及将其作为对抗训练数据集增强模型预测 CLR-DGA 域名的鲁棒性的可行性。

1.4 论文组织架构

根据本文的主要研究内容，将本文分为五部分，具体章节如下：

第一章为绪论。本章论述了本文的研究背景及意义，总结了国内外基于机器学习和深度学习的检测方法的研究现状，以及基于对抗攻击的新型 DGA 研究现

状,分析和对比了现有方法的优缺点,简要介绍了本文的主要贡献与创新、文章的结构内容与安排。

第二章为相关理论与技术原理。详细介绍了僵尸网络,域名系统,域名生成算法原理,以及本文提出的 DGA 域名检测方法所需的理论和技术。

第三章提出了基于深度学习的 DGA 域名分类模型。本章介绍了融合注意力机制与并行混合网络的 DGA 域名分类模型 PCBGA-DGA 的组成部分,介绍了实验环境、实验采用的数据集,通过对比实验与消融实验,验证了模型对良性域名及 20 种 DGA 家族的多分类性能,其中包含两种基于单词的 DGA。最后基于实验数据,进一步分析了部分难以检测的 DGA 家族的特点及它们之间的关系。

第四章是基于字符级替换的新型 DGA 域名生成及防御方法。本章提出了名为 CLR-DGA 的一种基于字符级替换的新型 DGA,并基于该 DGA 展开了对抗攻击和防御研究,观察 5 种深度学习分类器对包括 CLR-DGA 在内的 6 种 DGA 的检测效果。

第五章是总结与展望。本章对本文的工作内容进行了总结,并对未来的研究方向做出了展望。

第二章 相关理论及技术原理

2.1 僵尸网络

僵尸网络是一种由可接受相同指令的大量僵尸主机组成的逻辑网络,也是一种可被攻击者远程控制的通用攻击平台^[39],其规模从一千到数十万台计算机不等。1989年, Greg Lindahl 为使系统管理更加便利,基于 IRC (Internet relay chat) 协议架构撰写了 GM (Game manager for the hunt the wumpus game), 即僵尸网络的雏形。1999年, Pretty Park 发现了第一个使用 IRC 服务器作为远程控制服务器的恶意代码,这说明僵尸网络可以在 Client-Server 模式下运行,并为僵尸网络的大规模开发提供了契机。之后,僵尸网络受到了网络攻击者的青睐,并迅速地兴起与壮大。

借助僵尸网络可实现一对多控制的特性,攻击者可以高效地控制大量的计算机资源,并为 DDoS 攻击等恶意活动提供充分的条件。近年来,恶意软件即服务 (Malware-as-a-service) ^[40]正逐步成为趋势,通过这一付费服务,攻击者可以使用租用的僵尸网络实施恶意活动。这一趋势降低了执行恶意活动的门槛,吸引更多的脚本小子 (Script kiddie) 和犯罪团伙发动恶意攻击,也使得大规模的僵尸网络逐渐被小型即用的僵尸网络取代。具备一对多控制特性和向小型即用的方向发展的僵尸网络,再与 DGA、Fast-flux 等技术相结合,获得了更强的隐蔽性,给安全专家和系统管理员带来了巨大的挑战。深入理解僵尸网络以及 DGA 恶意域名的特征和发展趋势,有助于进一步优化针对僵尸网络的安全防御工作。

采用 DGA 的僵尸网络的运行机制如图 2.1 所示。从属于同一僵尸网络的 Bots 和 C&C 服务器可以使用同种 DGA 生成相同的候选 MAGD 列表。当攻击者即 Botmaster 需要与 Bots 建立通信时,只需从 MAGD 列表中选择较小的域名子集注册,将域名子集与 C&C 服务器使用的 IP 地址建立映射关系; Bots 的僵尸程序生成相同的 MAGD 列表,遍历每个 MAGD 并发起 DNS 查询请求,当某个 MAGD 对应的查询请求成功时, Bots 可与注册并使用该 MAGD 的 C&C 服务器成功建立通信信道。因此,及时检测来自 Bots 的 DNS 查询请求,有助于阻止僵尸网络的活动。

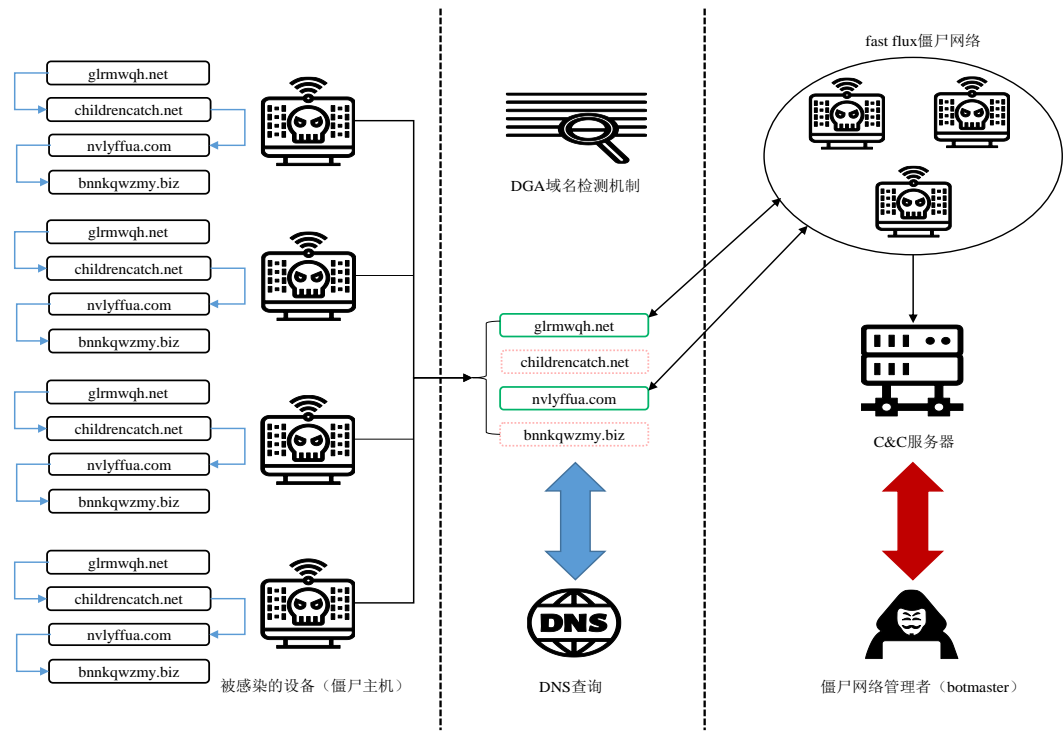


图 2.1 采用 DGA 的僵尸网络的运行机制

2.2 DGA 恶意域名

2.2.1 域名系统

主机的 IP 地址，是其在互联网中进行网络通信的唯一标识，然而 IP 地址完全由近似无规律的数字组成，难以记忆。为了使网络通讯更加便利，域名系统（Domain name system, DNS）^[41]正式面世。DNS 是互联网基础设施的重要组成部分，极大地推动了网络技术的普及和发展。DNS 将容易记忆的域名字符串自动解析为相应的 IP 地址，保证上层的网络应用能顺利找到目标主机。

域名，即完全限定域名（Fully qualified domain name, FQDN）^[42]，是由句点字符分隔的字符串序列，可以表示网络上某一台计算机或计算机组。一个 FQDN 可以分为顶级域名（Top-level domain, TLD）、二级域名（Second-level domain, SLD）、主机名等。例如，www.baidu.com 是一个 FQDN，其中 com 为 TLD，baidu 为二级域名，www.baidu.com 为 baidu.com 的子域名，同时 www.baidu.com 也被称作主机名，域名的各个部分被“.”分隔。DNS 协议规定，一个完整的域名字符串的总长度不能超过 253 个字符，被“.”分割的每一级域名的长度最长不超过 63 个字符，字符的选择范围是英文字母（不区分大小写）、数字和“-”共 37 种。由于易于记忆的需求，FQDN 的设计通常基于设计者使用的自然语言，因此 FQDN

通常具有作为英语或其它语言的词语组合的可读性。域名是创建网站的基础，也是直达网站的入口。对于企业来说，域名便于吸引流量，相当于企业商标，是重要的品牌资产，具有不可忽视的商业价值。许多企业不惜重金购买域名^[43]，例如 Facebook 的 fb.com、奇虎 360 的 360.com 等。

2.2.2 域名生成算法

域名生成算法(DGA)基于伪随机数生成器(Pseudo random number generator, PRNG)的原理，采用日期、元辅音、热门词汇等随机初始化种子随机生成一组字符串前缀，前缀后添加合适的 TLD 得到最终的伪随机域名集合。这些伪随机域名统称为 DGA 恶意域名或 MAGD。目前已知的 DGA 已有数十种。定义良好的 DGA 分类方法，不仅有利于充分了解 DGA 的用途、特性等，而且有助于开发新颖、有效的解决方案。

良性域名、恶意域名与 DGA 的关系如图 2.2 所示。DGA 可以分成 2 个子类：基于字符的 DGA 和基于单词的 DGA^[44]。其中基于字符的 DGA 又可以分成 3 个子类：基于算术的 DGA，基于哈希的 DGA，基于置换的 DGA。基于算术的 DGA，根据时间或者随机种子，生成一组可用 ASCII 编码表示的值，从而构成 MAGD，已知的 DGA 大部分都属于这一类，如 Conficker 家族的 obvbaoijs.com。基于哈希的 DGA，通常使用 MD5，SHA256 等哈希算法生成 MAGD，如 Bamital 家族的 00e3f9604c6dd1a3f08249de416dd201.org。基于置换的 DGA，对一个初始域名进行字符上的排列组合，目前已知的基于置换的 DGA 家族仅有 VolatileCedar 一种。基于单词的 DGA，会从专用的词典中选择单词并组合，减少域名字符串的随机性，迷惑性更强，例如 Matsnu 家族的 world-bite-care.com，以及 Suppobox 家族的 childrencatch.net。

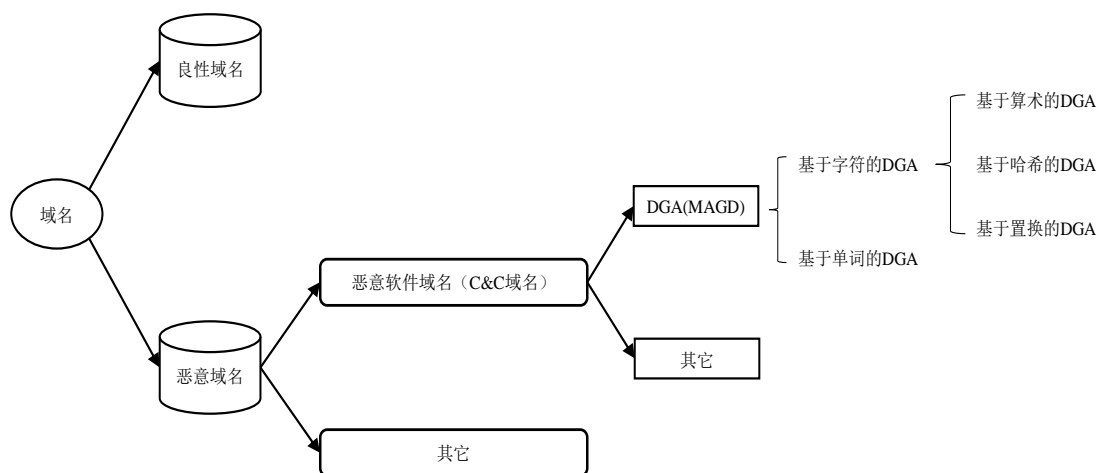


图 2.2 良性域名、恶意域名与 DGA 的关系

由上可知,基于字符的 DGA 以字符为单位组合而成,而基于单词的 DGA 通常以英文单词为单位组合而成。基于字符的 DGA 种类繁多,字符串随机性强,对该类 DGA 生成的 MAGD 的检测方法也相对最为成熟。基于单词的 DGA 种类屈指可数,但是其 MAGD 的构成更加逼近真实的良性域名,比大多数基于字符的 DGA 更难以检测。

2.3 深度学习

深度学习可以自动学习数据的内在特征,处理多种复杂的模式识别问题。深度学习解决了传统机器学习需要专业人士提取特征,人力成本及时间成本高的问题。在 DGA 恶意域名检测领域,Woodbridge 等人的 Endgame 模型^[23],证明了循环神经网络结构处理域名序列数据的巨大优势;而 TextCNN 模型的出现以及 Yu 等人的工作^[26],表明卷积神经网络也可对 DGA 域名文本进行分类并取得较好的效果。

2.3.1 卷积神经网络

卷积神经网络(CNN)^[45]是深度学习的一种经典算法。通常,随着层数的增加,神经网络模型的权重等参数量会快速膨胀,从而导致训练时间成本过高,而模型效果不佳;而 CNN 中的卷积结构可以显著减少训练模型时所需的内存空间,且大幅减少有待学习的参数量,进而缓解了模型的过拟合问题;同时在一般的前向传播算法的基础上提高了训练性能。实际应用时,CNN 可以通过不同维度的卷积层,处理不同维度的输入数据。由于域名数据是一维的文本输入,研究主要关注其序列在一维空间维度的关系,因而 DGA 恶意域名检测领域的 CNN 方法以一维卷积为主。CNN 包含稀疏连接、权值共享和池化这三个关键的操作,它们有助于减少 CNN 所要训练的参数,降低计算的复杂度。

(1) 稀疏连接:卷积层中的神经元仅与其相邻层的部分神经元相连,而不再和上一层的所有神经元相连,以显著减少网络中神经连接的数量。卷积层中不同的神经元只需感知输入数据中不同的局部区域,之后综合各神经元感知到的局部特征以获取输入数据的全局信息。

(2) 权值共享:卷积核(Kernel)内部定义了卷积计算所需的固定参数,在使用相同的卷积核遍历所有数据时,卷积计算的参数不变。

(3) 池化:类似于图像压缩的原理,通过降低数据维度,避免过拟合可能增强稀疏连接的问题。

一个完整的卷积神经网络如图 2.3 所示。卷积层、池化层以及全连接层交叉

堆叠，并通过反向传播算法进行训练。为了有效地提升模型性能，有时可增加卷积层或池化层的数量。

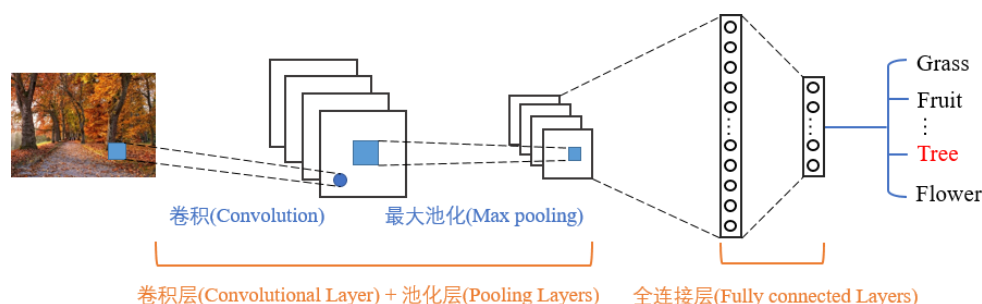


图 2.3 CNN 示意图

卷积层是 CNN 的核心，该层通过卷积核对输入数据执行卷积计算，抽取数据的局部特征，并输出相应的特征信息，这些特征信息也可被称为特征图。卷积核的数量决定了卷积操作后特征图的数量。CNN 模型包括若干个卷积层，每个卷积层可以含有若干个不同尺寸的卷积核，以提取不同的特征。

池化层又被称为取样层，该层的任务是选择并过滤卷积层提取到的特征信息，进行特征降维。常用的池化方式有最大池化和平均池化。在文本处理任务中，一般选择最大池化方式，有助于保留窗口内部的最大值特征，减少数据噪声的影响，进而增强模型的鲁棒性。最大池化还可以用定长输入取代变长输入，便于之后通过全连接层进一步处理。

全连接层的输入是训练数据经过若干次卷积、池化操作后得到的特征信息，其输出是一个 N 维向量的概率分布，然后该层通过 Softmax 函数计算域名是某种 DGA 的概率，并输出最大的概率对应的类别标签，完成分类工作。

2.3.2 循环神经网络

虽然 CNN 善于提取序列的局部特征，但是它无法学习序列之间的时间依赖关联，而循环神经网络（RNN）^[46]具有记忆性和参数共享的特性，适于学习对序列的时间依赖等非线性特征。近年来 RNN 在文本、语音及视频等序列数据的处理上表现出了很大的优势。RNN 可以处理可变长度的输入序列，捕捉上下文序列之间的语义关系，进而在模型训练时提取更加丰富的特征信息。

在处理域名数据时，RNN 可以有效地捕捉域名字符串的时序特征信息。但 RNN 在反向传播的过程中，后层的梯度会以连乘的方式叠加到前层，导致梯度消失或者梯度爆炸，限制了模型学习长序列的时间依赖关系的能力。为应对梯度消失和梯度爆炸问题，LSTM^[47]应运而生。LSTM 不仅保留了 RNN 的优点，还

支持更长的记忆功能，进而能够创建更深层的网络，以应对更加复杂的任务。LSTM 用记忆单元代替 RNN 中的隐藏层节点，记忆单元由遗忘门 f_t 、输入门 i_t 和输出门 o_t 组成。遗忘门决定了上一时刻的单元状态 C_{t-1} 有多少保留到当前时刻 C_t ，输入门决定了当前时刻网络的输入 x_t 有多少保存到单元状态 C_t ，输出门控制单元状态 C_t 有多少输出到 LSTM 的当前输出值 h_t 。LSTM 结构如图 2.4 所示。

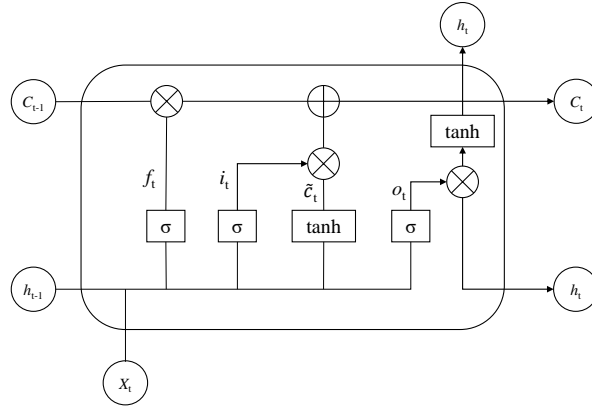


图 2.4 LSTM 示意图

之后，为了处理 LSTM 网络训练速度过慢的问题，Cho 等人^[48]提出了门控循环单元（Gated recurrent unit, GRU）这一变体。LSTM 中的遗忘门、输入门和输出门变更为 GRU 中的更新门和重置门。更新门用于控制前一时刻的信息被带入到当前状态中的程度，重置门用于控制前一状态中有多少信息被写入到当前时刻的候选集。在保留 LSTM 优点的基础上，GRU 结构更加简单，且参数量更少，模型训练更加方便快捷。图 2.5 为 GRU 内部结构示意图，其中 h_{t-1} 为 t-1 时刻隐藏层的输出， x_t 为 t 时刻输入， h_t 为 t 时刻隐藏层的输出，t 表示 t 时刻隐藏层节点候选值， r_t 表示重置门， z_t 表示更新门。其它元素操作如图例所示。

以公式描述图 2.5 如下。公式(2.1)~(2.4)中 W_r 、 W_z 、 W 为权重矩阵， σ 和 \tanh 分别为 sigmoid 激活函数和双切正切激活函数。

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (2.1)$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (2.2)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times h_t \quad (2.3)$$

$$h_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \quad (2.4)$$

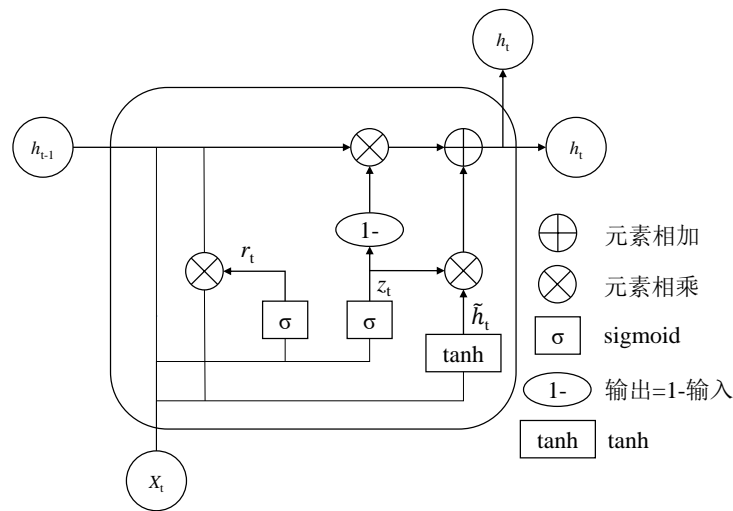


图 2.5 GRU 示意图

2.3.3 注意力机制

在深度学习领域，模型训练以接收并处理大量的数据为前提，但往往只有少量的数据会显著影响模型的训练效果。如何使模型提高对关键信息的利用率，成为了该领域的一个重要的研究课题。人类在观察事物时，会根据需要选择并关注视觉区域中特定的部分，同时忽略掉其它不相关的信息，这种机制有助于提高人类处理视觉信息的效率与准确性。在人类的视觉注意力的启发下，注意力机制（Attention mechanism）^[49]应运而生。注意力机制为输入信息分配不同的权重，使模型在巨量的数据中捕获对当前任务有益的关键信息，并忽略其它信息。注意力机制的本质如图 2.6 所示。

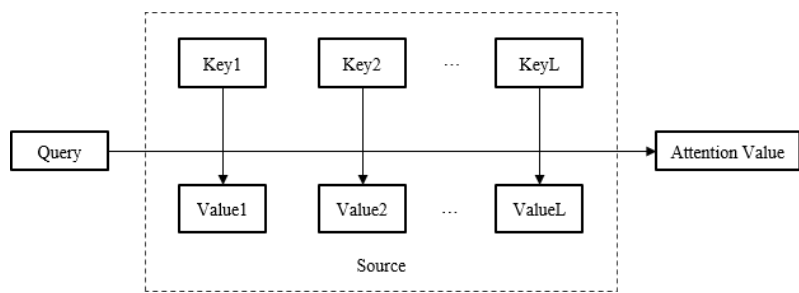


图 2.6 注意力机制的本质

可以将 Source 中的构成元素抽象为一系列 Key 和 Value 的数据对，此时只需给定 Target 中的某个元素 Query，就可以通过计算 Query 和各个 Key 之间的相关性，得到每个 Key 对应 Value 的权重系数，然后对 Value 加权求和，得到最终结果 attention value。本质上，注意力机制是对 Source 中元素的 Value 值进行加权求和，而 Query 和 Key 用于计算对应 Value 的权重系数，如公式(2.5)所示：

$$Attention(Query, Source) = \sum_{i=1}^L \text{Similarity}(Query, Key_i) \times Value_i \quad (2.5)$$

2.4 本章小结

本章首先介绍了僵尸网络、域名系统相关知识以及 DGA 的原理，然后介绍了基于深度学习的相关技术，列举并分析了 CNN、RNN 以及注意力机制等技术的特点，为后文的模型构建与实验做好铺垫。

第三章 基于深度学习的 DGA 域名分类模型

针对目前部分检测方法对域名信息利用率低，难以准确分类基于单词的 MAGD 等字符串随机性较低的 DGA 域名的问题，本文提出了一种基于深度学习的 DGA 域名分类模型（PCBGA-DGA）。该模型利用并行卷积神经网络（parallel CNN, PCNN）和结合注意力机制的双向门控循环单元（Bidirectional GRU, BiGRU），分别提取域名序列的局部特征和上下文的时序依赖特征；之后融合这两种特征信息，使用深度学习模型对公开的域名数据集开展多分类实验。同时，本文进一步分析了各模型对部分 DGA 家族分类效果较差可能的原因。实验结果表明，与传统的深度学习方法相比，PCBGA-DGA 模型在多分类 MAGD 时取得了更好的效果。

3.1 PCBGA-DGA 模型架构

本文的 MAGD 检测集成模型 PCBGA-DGA 总体框架如图 3.1 所示，其中 m 指代 DGA 类别数量。该框架包含域名数据预处理、嵌入层、特征提取层、输出层 4 个部分。

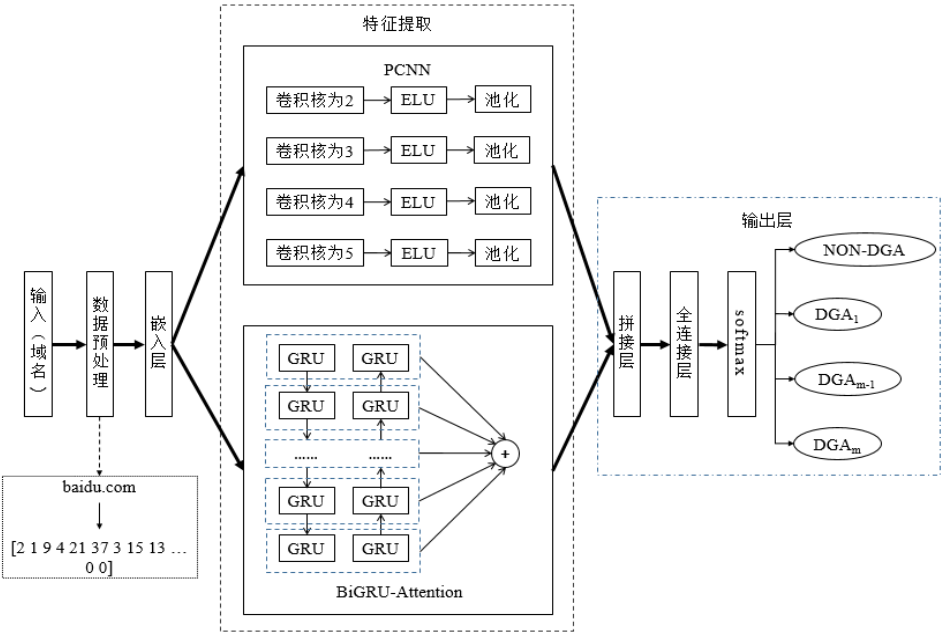


图 3.1 PCBGA-DGA 的总体框架

3.1.1 预处理

本模型的预处理步骤有两步：其一，将域名进行分词处理，将构成域名的各个字符替换为数字。其二，通过填充 0，统一域名的长度。

首先，分词用数字替换域名的每个字符，以使深度学习模型可处理域名数据。本章使用 Keras (<http://keras.io/>) 的 Tokenizer 将字母 (a 到 z)、数字 (0 到 9) 和特殊字符 (“-”, “.”) 替换为整数 1 到 38，整数 0 用于填充。根据域名的生成条件，英文字母不区分大小写。

其次，填充使域名向量的长度统一。如图 3.2 所示，实验涉及的所有域名的长度范围从 4 到 66。本研究在所有长度不足 66 的域名向量左侧填充 0，直到长度统一为最大长度 66，以尽可能充分地提取域名的信息。

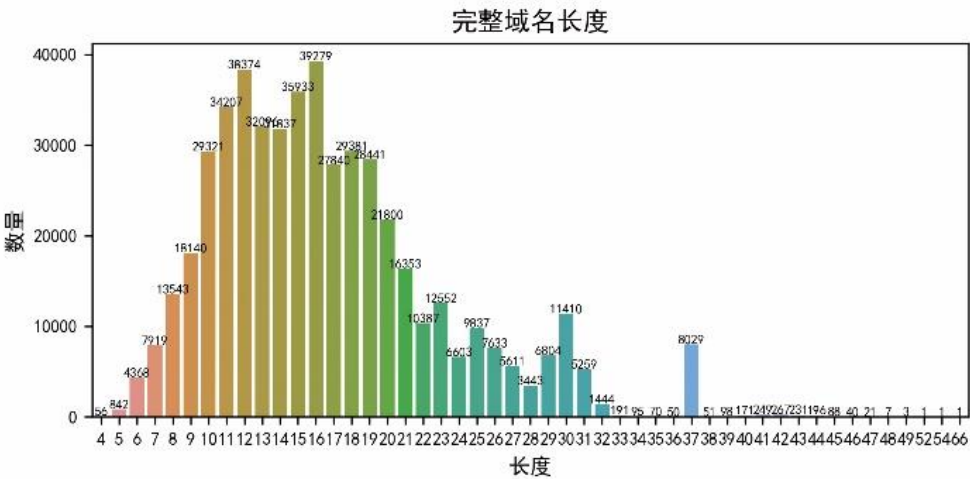


图 3.2 域名的长度分布

3.1.2 嵌入层

嵌入层将预处理后的域名映射为密集向量，密集向量在深度学习模型中进行更新。近年来，在自然语言处理中，单词级嵌入表现出了比字符级嵌入更优的性能。然而，域名是一种字符序列，总长度较短，而且不完全由单词组成。因此，对于域名，使用字符级嵌入效果更佳。

本研究使用的字符级嵌入的相关参数如下。将嵌入层的输入大小设置为 66，并将输出维度设置为 32。对于字符级嵌入，32 的输出维度可在计算的精度和复杂性之间提供一个很好的折衷^[27]。为防止过拟合，本层还应用了随机失活 (Dropout) 和 L2 正则化方法。每个域名经过嵌入层后，可表示为 66×32 维向量。

3.1.3 特征提取层

本文提出了一种混合装袋和堆叠的 DGA 域名特征提取方法,该方法通过全连接层整合各模型的中间值。使用 PCNN 和结合注意力机制的 BiGRU 作为子模型,既利用了可学习局部序列信息的 CNN,也采用了学习全局序列信息的 BiGRU,还包含了可学习权重信息的注意力机制。所提出的集成模型使用每个子模型的中间值作为全连接层的输入,而不是以端到端的方式训练子模型的传统集成方法。以这种方式,集成模型可以更好地保留两个子模型提取的特征信息,使分类结果更加精确。该混合方法分为两部分:PCNN 模型和 BiGRU-Att 模型。

3.1.3.1 PCNN 模型

CNN 通常用于图像数据处理,但它也可以处理一维序列数据,这时的 CNN 称为 1D-CNN。1D-CNN 具有能够学习局部信息、可以并行计算、训练速度快的优势,主要用于传感器数据时间序列分析或自然语言处理。域名是长度有限的字符序列,在预处理统一长度后,可使用 1D-CNN 进行分类。

本文使用 PCNN 模型^[27],该模型通过多种尺寸的内核提取域名序列的局部特征。模型结构和参数参考现有的基于并行 CNN 模型的方法^[26]。首先,在特征提取步骤中,应用四个并行的一维卷积层,卷积核大小从 2 到 5 不等,各卷积层的过滤器(Filter)参数,即卷积核数量取值均为 256。在每个一维卷积层之后应用最大池化层和 Dropout 层,以降低节点间的相互依赖性并防止过拟合。

3.1.3.2 BiGRU-Att 模型

早期的 RNN 模型在模型训练中更注重序列信息末尾的语义信息,即存在信息偏倚的缺陷。为应对信息偏倚问题,Schuster 等人^[50]在单向 RNN 的基础上添加了一个反方向的隐藏层,生成了一种双向循环神经网络(Bidirectional RNN, BiRNN),让模型可以学习到上下文之间存在的双向序列信息。在 RNN 基础上改进的 LSTM、GRU 模型则有助于应对梯度消失和梯度爆炸问题。与 LSTM 相比,GRU 可以更加有效地减少过拟合风险。随着深度学习技术的不断发展,结合 BiRNN 与 GRU 的 BiGRU 模型已可用于文本分类^[51]。

注意力机制本质上是为特征赋予权重,该权重由模型自行计算,通过权重分配,让模型更好地学习核心的特征信息。2017 年,谷歌的翻译团队在机器翻译任务中使用自注意力(Self-attention)机制^[52]代替传统的 RNN 模型,提升了模型处理的效果。与传统的注意力机制相比,自注意力机制大幅降低了对外部信息的依赖程度,在文本分类任务中捕捉数据的内部相关性更有优势。Qiao 等人^[53]使用

结合简化自注意力机制的 LSTM 模型检测 DGA 域名，并发现在分类 DGA 域名时，域名中每个字符的重要性是不同的。

BiGRU 包括两个方向相反的 GRU，两者参数彼此独立，输入的序列向量经过 BiGRU 后，得到双向的输出，再链接输出即可得到特征集合 H ，如公式(3.1)所示：

$$H = \text{concatenate}(\overrightarrow{H}, \overleftarrow{H}) \quad (3.1)$$

再借助注意力机制进一步处理 BiGRU 提取到的序列特征信息，增加核心序列特征的占比。BiGRU 层的输出 $H = [h_1, \dots, h_l]^T$ ，表示长度为 l 的域名序列经过 BiGRU 后得到的 l 个特征向量，其中 $h_i = [h_i^1, \dots, h_i^g, \dots, h_i^{2g}]$ ， g 是 GRU 的神经单元个数， $2g$ 代表前向与反向的输出特征总数。在本章的实验中， g 为 128。此时 H 也可表示为公式(3.2)：

$$H = \begin{pmatrix} h_1^1 & \cdots & h_1^{2g} \\ \vdots & \ddots & \vdots \\ h_l^1 & \cdots & h_l^{2g} \end{pmatrix} \quad (3.2)$$

由公式(3.2)可知， H 的每一列 $[h_i^1, \dots, h_i^{2g}]^T$ 是长度为 l 的完整特征组合。对 H^T 的每一行进行 Softmax 函数运算，得到 H 的注意力权重矩阵 S^T ，如公式(3.3)所示：

$$S^T = \begin{pmatrix} \frac{e^{h_1^1}}{\sum_{j=1}^l e^{h_j^1}} & \cdots & \frac{e^{h_l^1}}{\sum_{j=1}^l e^{h_j^1}} \\ \vdots & \ddots & \vdots \\ \frac{e^{h_1^g}}{\sum_{j=1}^l e^{h_j^g}} & \cdots & \frac{e^{h_l^g}}{\sum_{j=1}^l e^{h_j^g}} \\ \vdots & \ddots & \vdots \\ \frac{e^{h_1^{2g}}}{\sum_{j=1}^l e^{h_j^{2g}}} & \cdots & \frac{e^{h_l^{2g}}}{\sum_{j=1}^l e^{h_j^{2g}}} \end{pmatrix} = \begin{pmatrix} \alpha_1^1 & \cdots & \alpha_l^1 \\ \vdots & \ddots & \vdots \\ \alpha_1^g & \cdots & \alpha_l^g \\ \vdots & \ddots & \vdots \\ \alpha_1^{2g} & \cdots & \alpha_l^{2g} \end{pmatrix} \quad (3.3)$$

最后，将 H^T 与注意力权重矩阵 S^T 进行点乘 \odot ，得到注意力层的输出 O^T ，如公式(3.4)所示：

$$O^T = H^T \odot S^T = \begin{pmatrix} h_1^1 \alpha_1^1 & \cdots & h_l^1 \alpha_l^1 \\ \vdots & & \vdots \\ h_1^g \alpha_1^g & \ddots & h_l^g \alpha_l^g \\ \vdots & & \vdots \\ h_1^{2g} \alpha_1^{2g} & \cdots & h_l^{2g} \alpha_l^{2g} \end{pmatrix} \quad (3.4)$$

3.1.4 输出层

最后，将特征提取层的两个子模型的输出拼接，再通过一个全连接层对 DGA 域名进行分类。全连接层的神经元个数等于要分类的 DGA 类别的数目，其中良性域名即非 DGA 域名，也认为是一种 DGA 类别。使用多分类问题中广泛应用的 Softmax 函数作为激活函数。Softmax 函数如公式(3.5)所示：

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (3.5)$$

其中， n 为分类的类别数， x_i 为全连接层第 i 个神经元输入的信息， $\text{Softmax}(x_i)$ 为全连接层第 i 个神经元输出的信息，对应当前域名是第 i 类域名的概率。Softmax 对同一域名输出的所有概率值之和为 1。

3.2 实验与分析

3.2.1 实验环境

本章的具体实验环境和硬件配置如表 3.1 所示。

表 3.1 实验环境具体配置

环境配置	参数
操作系统	Ubuntu 18.04
内存	15 GB
CPU	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
GPU	GTX 1080 Ti 11GB
Anaconda3	4.10.3 版本 64 位
Python	3.6
Tensorflow	1.14

3.2.2 数据集和评估指标

本章实验数据集的分布情况如表 3.2 所示，其中包括 250,000 个非 DGA 和 250,573 个 DGA 域名。

表 3.2 实验数据集分布

DGA 编号	DGA 家族	数量
0	Non-DGA(Tranco)	250,000
1	Banjori	50,000
2	Tinba	20,000
3	Post	20,000
4	Ramnit	20,000
5	Qakbot	20,000
6	Necurs	20,000
7	Murofet	20,000
8	Urlzone	10,000
9	Simda	10,000
10	Ranbyus	10,000
11	Pykspa	10,000
12	Dyre	7,998
13	Kraken	7,878
14	Cryptolocker	6,000
15	Nymaim	6,000
16	Locky	4,014
17	Vawtrak	3,150
18	Shifu	2,331
19	Suppobox	2,297
20	Matsnu	905
DGA 总数		250,573
域名总数		500,573

250,573 个 DGA 域名取自 Bambenek Consulting OSINT (<https://osint.bambenekconsulting.com/feeds/>) 和 Netlab DGA (<https://data.netlab.360.com/dga/>)。Bambenek Consulting OSINT 数据集由网络安全和威胁情报咨询公司 Bambenek C

onsulting 提供, 包含 Banjori 和 Tinba 等 50 多种 DGA。本章从中选取数量排名前 18 名的 DGA, 这些都是基于字符的 DGA。由于 Bambenek 数据集中, Suppo box 和 Matsnu 这两种典型的基于单词的 DGA 的数量过少, 故这两个家族从 Net lab DGA 收集。基于单词的 DGA 虽然种类与数量均较少, 但其生成的域名更加接近非 DGA 域名, 相比基于字符的 DGA 有更大的威胁, 因此为更准确地评估模型性能, 有必要引入基于单词的 DGA。为接近真实世界良性域名和 DGA 域名的分布, Bambenek Consulting OSINT 中 MAGD 数量超过 10 万的 Banjori 随机选取 5 万个, MAGD 数量在 2 万到 10 万之间的 Tinba 等 6 种 DGA 分别选取 2 万个, MAGD 数量在 1 万到 2 万之间的 Urlzone 等 4 种 DGA 分别选取 1 万个, 其余 9 种数量不足 1 万个的 DGA 的 MAGD 全部选取。以上共 20 种 DGA 生成的域名标记为类别 1 到 20。

250,000 个非 DGA 域名选自 Tranco(<https://tranco-list.eu/list/8KYV/1000000>) 排名前 25 万的部分。之前的相关研究主要选择 Alexa (<http://alexa.com/topsites/>) 作为良性域名数据源。然而, 已有工作表明 Alexa 列表排名前 3 万的域名中也有可能包含恶意域名^[54]。因此, 本文使用综合分析包含 Alexa 在内共 4 个数据源并每日更新的流行域名排行列表 Tranco, 作为本工作中良性域名的来源。与 Alexa 相比, Tranco 中 DGA 域名名列前茅的难度更高。来自 Tranco 的非 DGA 域名标记为类别 0。本章使用交叉验证技术公平地评估各深度学习分类模型, 将 500,573 个域名按照 8:1:1 的比例划分为训练集、验证集和测试集。

本章使用精确度 (Precision)、召回率 (Recall) 和 F1 分数 (F1-score) 来衡量模型的分类性能, 3 个指标的计算由公式(3.6)~(3.8)确定。精确度表示模型的可信度; 召回率反映模型的漏报情况; F1 值综合了精确度和召回率两种指标, 可以较好地体现模型的整体分类性能。

$$Precision = \frac{TP}{TP + FP} \quad (3.6)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.7)$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3.8)$$

本文还比较了深度学习模型多分类时各指标的宏平均和加权平均结果。宏平均默认各类别权重相同, 而加权平均则根据各类别数据占比分配权重。宏平均适合度量每个类的平均性能, 而加权平均更能反映整个数据集的性能。在多分类不平衡数据集的场景下, 加权平均通常比宏平均更公平。因此, 为了更全面地衡量数据集和模型的性能, 实验比较了所有模型各指标的宏平均和加权平均的结果。

3.2.3 多分类对比实验

为了验证本章方法的有效性, 设置了 Endgame^[23], LSTM-Att^[53], Bilbo^[55]、CL^[56]四个对比模型, 其中 Endgame、LSTM-Att 为基于 RNN 结构的模型, Bilbo 和 CL 为结合 CNN 和 RNN 而成的混合模型。

(1) Endgame 模型^[23], 使用包含 128 个神经元的单向 LSTM, 提取域名序列的长依赖关系, 最后使用全连接层分类域名。

(2) LSTM-Att 模型^[53], 在 LSTM 提取依赖关系的基础上, 结合自注意力机制提取字符权重信息。

(3) Bilbo 模型^[55], 其特征提取模块由浅层 CNN 和单向 LSTM 并行组成, 最后通过全连接层输出分类结果。该模型中包含卷积核大小为 2 到 6 的 5 个 CNN 结构, 每个 CNN 的卷积核数量为 128; 每个 LSTM 的记忆单元数量为 128。

(4) CL 模型^[56], 先用卷积核大小为 3 和 4 的两个并行的 CNN 结构学习域名序列的局部特征, 之后每个 CNN 的输出分别对接一个单向 LSTM, 以提取域名序列的长依赖关系, 最后通过全连接层输出分类结果。该模型中每个 CNN 的卷积核数量为 128, 每个 LSTM 的记忆单元数量为 128。

所有深度学习模型统一使用如下超参数: Dropout 层丢失率为 0.5, 训练 10 个迭代, 学习率 (Learning rate) 为 $1e-4$, 批处理大小 (Batch size) 为 64, Adam 优化器作为优化函数。

在相同的实验环境、数据集和超参数的配置下, 本文方法与四个模型进行对比, 各模型对不同 DGA 家族的多分类性能比较如表 3.3 所示。

就总体分类效果而言, 在对比实验的 5 个模型中, 本文提出的 PCBGA-DGA 集成模型的精确度、召回率、F1 分数三个指标的宏平均及加权平均结果均为最高。与次优的 LSTM-Att 模型相比, 集成模型的宏平均 F1 分数高出近 5.5 个百分点, 加权平均 F1 分数高出 1.7 个百分点。次优模型为 LSTM 结合自注意力机制, 而非其它无自注意力机制的模型, 说明与不含自注意力机制的单一 RNN 结构或 RNN 与 CNN 集成的模型相比, 结合自注意力机制的 RNN 结构可以更加有效地学习 DGA 域名内部的特征, 进而得到更好的 DGA 域名多分类效果。在所有模型中, 基于单层 LSTM 结构的 Endgame 模型分类效果最低, 说明 Bilbo 和 CL 等模型在不含注意力机制的情况下, 集成 RNN 与 CNN 也有助于小幅提升对 DGA 域名的总体分类性能。

与其它模型相比, 集成模型对 Murofet、Urlzone、Cryptolocker、Post 这四类基于字符的 DGA 分类效果未能达到最优。LSTM-Att 分类 Murofet、Cryptolocker 的 F1 分数最高, Murofet 高于集成模型 3 个百分点以上, Cryptolocker 则高于集成模型 6.5 个百分点以上。Bilbo 模型分类 Urlzone、Post 的 F1 分数最高, Urlzone

高于集成模型约 2 个百分点, Post 则高于集成模型 0.02 个百分点。同时, 集成模型对剩下的包含良性域名在内的 17 种域名分类效果均为最高。尤其是分类训练和测试数据最少的 Suppobox 和 Matsnu 两个基于单词的 DGA 家族时, 与次优的 LSTM-Att 相比, 集成模型分类 Suppobox 的 F1 分数提高了 16 个百分点, 分类 Matsnu 的 F1 分数提高了近 7.5 个百分点。这说明本章的集成模型对包含基于单词的 DGA 在内的多数 DGA 的分类性能相比其它模型有更大的优势。

表 3.3 多分类对比实验

DGA	Endgame			LSTM-Att			Bilbo			Sup-port
	P	R	F1	P	R	F1	P	R	F1	
Tranco	0.9500	0.9786	0.9641	0.9643	0.9813	0.9727	0.9514	0.9932	0.9718	24710
Banjori	0.9760	0.9992	0.9875	0.9955	0.9996	0.9976	0.9969	1.0000	0.9984	5127
Tinba	0.8020	0.9732	0.8793	0.8684	0.9839	0.9225	0.7828	0.9747	0.8682	2052
Post	0.9971	0.9995	0.9983	1.0000	0.9990	0.9995	1.0000	1.0000	1.0000	2056
Ramnit	0.6613	0.7796	0.7156	0.6993	0.8527	0.7684	0.6722	0.8252	0.7409	1996
Qakbot	0.6890	0.6673	0.6780	0.7777	0.6830	0.7273	0.7262	0.6222	0.6702	2038
Necurs	0.7300	0.7223	0.7261	0.8927	0.7423	0.8106	0.8309	0.7792	0.8042	2006
Murofet	0.7928	0.8535	0.8221	0.8653	0.8751	0.8702	0.7514	0.8371	0.7919	1946
Urlzone	0.9565	0.9064	0.9308	0.9888	0.8903	0.9370	0.9868	0.9024	0.9427	994
Simda	0.9420	0.9448	0.9434	0.9351	0.9941	0.9637	0.8993	0.9852	0.9403	1015
Ranbyus	0.6924	0.6634	0.6776	0.7511	0.8249	0.7863	0.6779	0.6654	0.6716	1028
Pykspa	0.8476	0.8428	0.8452	0.9626	0.9367	0.9495	0.9617	0.8907	0.9248	1043
Dyre	0.9988	1.0000	0.9994	0.9950	1.0000	0.9975	0.9975	1.0000	0.9988	801
Kraken	0.9833	0.7449	0.8477	0.9327	0.8220	0.8738	0.9983	0.7626	0.8647	792
Cryptolocker	0.5444	0.3316	0.4121	0.5557	0.5829	0.5690	0.6424	0.1693	0.2680	573
Nymaim	0.3627	0.1645	0.2264	0.4911	0.3962	0.4385	0.3908	0.2173	0.2793	626
Locky	1.0000	0.0024	0.0049	0.7589	0.2616	0.3891	0.6667	0.0342	0.0651	409
Vawtrak	0.0000	0.0000	0.0000	0.7805	0.3097	0.4434	0.0000	0.0000	0.0000	310
Shifu	0.7107	0.9113	0.7986	0.8674	0.9758	0.9184	0.6558	0.8911	0.7556	248
Suppobox	0.7541	0.2266	0.3485	0.6415	0.6700	0.6554	0.0000	0.0000	0.0000	203
Matsnu	0.0000	0.0000	0.0000	0.4634	0.2235	0.3016	0.0000	0.0000	0.0000	85
宏平均	0.7329	0.6530	0.6574	0.8184	0.7621	0.7758	0.6947	0.6452	0.6455	50058
加权平均	0.8843	0.8942	0.8828	0.9195	0.9207	0.9173	0.8882	0.9033	0.8898	50058

表 3.3 (续)

DGA	CL			PCBGA-DGA			Sup-port
	P	R	F1	P	R	F1	
Tranco	0.9510	0.9911	0.9707	0.9788	0.9907	0.9847	24710
Banjori	0.9948	0.9998	0.9973	0.9986	1.0000	0.9993	5127
Tinba	0.8368	0.9742	0.9002	0.8603	0.9966	0.9235	2052
Post	0.9990	0.9985	0.9988	0.9995	1.0000	0.9998	2056
Ramnit	0.6980	0.8382	0.7617	0.7664	0.7856	0.7759	1996
Qakbot	0.6860	0.7149	0.7001	0.8148	0.6820	0.7425	2038
Necurs	0.9179	0.7752	0.8405	0.9529	0.8161	0.8792	2006
Murofet	0.8652	0.8309	0.8477	0.7605	0.9368	0.8395	1946
Urlzone	0.9717	0.8974	0.9331	0.9235	0.9235	0.9235	994
Simda	0.9601	0.9714	0.9657	0.9720	0.9931	0.9825	1015
Ranbyus	0.7386	0.7860	0.7615	0.8031	0.8969	0.8474	1028
Pykspa	0.9305	0.9492	0.9397	0.9550	0.9962	0.9751	1043
Dyre	0.9988	1.0000	0.9994	0.9988	1.0000	0.9994	801
Kraken	0.9625	0.8093	0.8793	0.9891	0.8018	0.8856	792
Cryptolocker	0.6048	0.3525	0.4454	0.6237	0.4223	0.5036	573
Nymaim	0.4031	0.2061	0.2727	0.7159	0.5192	0.6019	626
Locky	0.8857	0.2274	0.3619	0.6478	0.4768	0.5493	409
Vawtrak	1.0000	0.0194	0.0380	0.9029	0.9000	0.9015	310
Shifu	0.7726	0.8629	0.8152	0.9060	0.9718	0.9377	248
Suppobox	1.0000	0.0542	0.1028	0.9539	0.7143	0.8169	203
Matsnu	0.0000	0.0000	0.0000	0.6875	0.2588	0.3761	85
宏平均	0.8179	0.6790	0.6920	0.8672	0.8134	0.8307	50058
加权平均	0.9107	0.9138	0.9041	0.9358	0.9369	0.9343	50058

在分类样本数量最少的 Matsnu 时, Endgame、Bilbo、CL 模型的所有指标均为 0, 而含有自注意力机制的 LSTM-Att 和集成模型有一定的分类效果, 这说明相比单一的 RNN、CNN 模型或它们的集成, 自注意力机制面对小样本学习的场景有显著的优势。并且, 集成模型分类非 DGA 域名的 F1 分数也是最高, 可达 0.9847。对非 DGA 域名的准确分类有助于在实际检测 DGA 域名时减少对正常网站服务的干扰。

3.2.4 多分类消融实验

本节的消融实验将 PCBGA-DGA 消融为 PCNN、GRU、BiGRU、GRU-Att、BiGRU-Att，比较各模型之间的多分类效果。实验结果如表 3.4 所示。

表 3.4 多分类消融实验

DGA	GRU			BiGRU			GRU-Att			Sup-port
	P	R	F1	P	R	F1	P	R	F1	
Tranco	0.9540	0.9788	0.9662	0.9385	0.9887	0.9630	0.9592	0.9817	0.9703	24710
Banjori	0.9809	0.9990	0.9899	0.9829	0.9990	0.9909	0.9909	0.9996	0.9952	5127
Tinba	0.8122	0.9820	0.8890	0.8160	0.9576	0.8812	0.8487	0.9815	0.9103	2052
Post	0.9985	0.9990	0.9988	0.9995	0.9981	0.9988	0.9961	1.0000	0.9981	2056
Ramnit	0.6515	0.7971	0.7170	0.6461	0.7921	0.7117	0.7256	0.8307	0.7746	1996
Qakbot	0.6879	0.6089	0.6460	0.6814	0.6413	0.6608	0.7141	0.7390	0.7263	2038
Necurs	0.8213	0.7403	0.7787	0.8185	0.7104	0.7606	0.8905	0.7463	0.8120	2006
Murofet	0.7375	0.8690	0.7978	0.7955	0.7816	0.7885	0.8348	0.9034	0.8677	1946
Urlzone	0.9541	0.8994	0.9259	0.9650	0.8863	0.9240	0.9823	0.8944	0.9363	994
Simda	0.9269	0.9862	0.9556	0.9443	0.9695	0.9567	0.9243	0.9744	0.9487	1015
Ranbyus	0.7188	0.7558	0.7368	0.7360	0.7782	0.7565	0.7577	0.8580	0.8047	1028
Pykspa	0.8529	0.9281	0.8889	0.8674	0.8658	0.8666	0.9184	0.9712	0.9441	1043
Dyre	0.9963	1.0000	0.9981	0.9975	1.0000	0.9988	1.0000	1.0000	1.0000	801
Kraken	0.9933	0.7500	0.8547	0.9900	0.7525	0.8551	0.9889	0.7866	0.8762	792
Cryptolocker	0.6227	0.2967	0.4019	0.6332	0.2862	0.3942	0.5871	0.4939	0.5365	573
Nymaim	0.3394	0.1486	0.2067	0.3333	0.1118	0.1675	0.5276	0.1677	0.2545	626
Locky	0.6813	0.1516	0.2480	0.9388	0.1125	0.2009	0.7244	0.2763	0.4000	409
Vawtrak	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.8378	0.2000	0.3229	310
Shifu	0.6962	0.8871	0.7801	0.6865	0.8831	0.7725	0.9027	0.9355	0.9188	248
Suppobox	0.6364	0.0345	0.0654	0.6364	0.0345	0.0654	0.7603	0.4532	0.5679	203
Matsnu	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.6875	0.1294	0.2178	85
宏平均	0.7172	0.6577	0.6593	0.7337	0.6452	0.6530	0.8361	0.7297	0.7516	50058
加权平均	0.8862	0.8984	0.8872	0.8839	0.8965	0.8836	0.9146	0.9183	0.9117	50058

表 3.4 (续)

DGA	PCNN			BiGRU-Att			PCBGA-DGA			Sup-port
	P	R	F1	P	R	F1	P	R	F1	
Tranco	0.9275	0.9954	0.9602	0.9676	0.9827	0.9751	0.9788	0.9907	0.9847	24710
Banjori	0.9969	0.9994	0.9981	0.9984	1.0000	0.9992	0.9986	1.0000	0.9993	5127
Tinba	0.5439	0.8869	0.6743	0.8842	0.9825	0.9307	0.8603	0.9966	0.9235	2052
Post	0.9980	0.9713	0.9845	0.9985	1.0000	0.9993	0.9995	1.0000	0.9998	2056
Ramnit	0.6814	0.6473	0.6639	0.7406	0.8126	0.7750	0.7664	0.7856	0.7759	1996
Qakbot	0.6773	0.5437	0.6032	0.7928	0.6742	0.7287	0.8148	0.6820	0.7425	2038
Necurs	0.7274	0.7582	0.7425	0.9387	0.8016	0.8647	0.9529	0.8161	0.8792	2006
Murofet	0.7674	0.7189	0.7424	0.8060	0.9265	0.8621	0.7605	0.9368	0.8395	1946
Urlzone	0.9299	0.8410	0.8833	0.9435	0.9074	0.9251	0.9235	0.9235	0.9235	994
Simda	0.9243	0.9389	0.9316	0.9325	0.9931	0.9618	0.9720	0.9931	0.9825	1015
Ranbyus	0.5797	0.1663	0.2585	0.8207	0.8862	0.8522	0.8031	0.8969	0.8474	1028
Pykspa	0.9028	0.8821	0.8923	0.9393	0.9789	0.9587	0.9550	0.9962	0.9751	1043
Dyre	0.9988	1.0000	0.9994	1.0000	1.0000	1.0000	0.9988	1.0000	0.9994	801
Kraken	1.0000	0.7614	0.8645	0.9377	0.8548	0.8943	0.9891	0.8018	0.8856	792
Cryptolocker	0.6351	0.1640	0.2607	0.6133	0.5148	0.5598	0.6237	0.4223	0.5036	573
Nymaim	0.3792	0.2508	0.3019	0.5115	0.4265	0.4652	0.7159	0.5192	0.6019	626
Locky	0.6250	0.0122	0.0240	0.7943	0.3399	0.4760	0.6478	0.4768	0.5493	409
Vawtrak	0.5000	0.0129	0.0252	0.7931	0.4452	0.5702	0.9029	0.9000	0.9015	310
Shifu	0.5553	0.8306	0.6656	0.9023	0.9677	0.9339	0.9060	0.9718	0.9377	248
Suppobox	0.6250	0.0246	0.0474	0.5931	0.6749	0.6313	0.9539	0.7143	0.8169	203
Matsnu	0.0000	0.0000	0.0000	0.5556	0.0588	0.1064	0.6875	0.2588	0.3761	85
宏平均	0.7131	0.5908	0.5964	0.8316	0.7728	0.7843	0.8672	0.8134	0.8307	50058
加权平均	0.8622	0.8712	0.8546	0.9254	0.9275	0.9241	0.9358	0.9369	0.9343	50058

由表 3.4 可知, 对于集成模型而言, BiGRU-Att 模型对总体分类性能提升的贡献优于 PCNN。与 PCNN 相比, BiGRU-Att 的所有指标, 加权平均时高出约 5 到 7 个百分点, 宏平均时高出约 12 到 19 个百分点。且 BiGRU-Att 对 Murofet, Urlzone, Tinba, Ranbyus, Dyre, Kraken, Cryptolocker 这 7 类基于字符的 DGA 的分类效果略胜于集成模型。对于 BiGRU-Att 而言, 自注意力机制比 RNN 结构双向处理信息对分类性能的提升有更大的贡献、与 BiGRU 相比, GRU-Att 的所有指标, 加权平均时高出约 2 到 3 个百分点, 宏平均时高出约 8 到 10 个百分点。

且 GRU-Att 对 Murofet, Urlzone, Matsnu 的分类效果略胜于 BiGRU-Att。PCNN 与 BiGRU 对 Suppobox 和 Matsnu 分类时, 宏平均及加权平均 F1 分数在 0 到 0.0654 之间, 与 GRU-Att 和 BiGRU-Att 相比低了数十个百分点, 这说明自注意力机制可使检测模型更多地关注有效的特征信息, 进而提高对基于单词的 DGA 的分类精度。

分类 Murofet 和 Urlzone 时, GRU-Att 模型的效果最优, F1 分数分别高出集成模型近 3 个百分点和 1.3 个百分点。分类 Matsnu 时, GRU-Att 的 F1 分数比 BiGRU-Att 高出 11 个百分点, 而比集成模型低了约 16 个百分点。推测以上三个家族构成的 DGA 域名, 正向的时序依赖信息随机性低于反向的时序依赖信息, 致使在同样有自注意力机制的前提下, BiGRU 分类效果不如 GRU。

分类 Tinba, Ranbyus, Dyre, Kraken, Cryptolocker 这 5 个家族时, BiGRU-Att 的效果最优。就 F1 分数而言, BiGRU-Att 与集成模型相比, Tinba 高出 0.7 个百分点, Ranbyus 高出 0.5 个百分点, Dyre 高出 0.06 个百分点, Kraken 高出 0.9 个百分点, Cryptolocker 高出 5.5 个百分点。而集成模型对包含良性域名在内的 14 种域名分类效果均为最优。尤其是分类训练和测试数据最少, 且检测难度较高的 Suppobox 和 Matsnu 两个基于单词的 DGA 家族时, 集成模型分类 Suppobox 的 F1 分数比次优的 BiGRU-Att 提高了 18.5 个百分点, 集成模型分类 Matsnu 的 F1 分数比次优的 GRU-Att 提高了近 16 个百分点。集成 PCNN 与 BiGRU-Att, 与 BiGRU-Att 相比, 虽然降低了少数基于字符的 DGA 的分类效果, 但大幅提升了对样本数量少且分类难度高的 Suppobox 和 Matsnu 的分类性能, 且对其它家族的检测效果也有一定的提升, 因此集成模型取得了综合最优的分类性能。

3.2.5 DGA 家族关联分析

通过混淆矩阵图, 可以更加直观地进行分析呈现多分类结果, 如图 3.3 所示。图 3.3 中, 行表示域名的实际类别, 列表示深度学习模型预测的类别, 图中的色块表示水平轴上的域名类别被分类为垂直轴上的域名类别比例, 其中 0 为白色, 1 为深红色。

可以发现, 有接近 29% 的 Suppobox 和超过 74% 的 Matsnu 的 MAGD 没有被正确分类, 而是被分类为 Tranco 域名。这两类基于单词的 DGA, 除了训练样本占比最少之外, 其 MAGD 字符串构成也与良性域名相近。Suppobox 每隔 512 秒将生成一个由单词表中的两个单词组成的 MAGD, 并采用固定的 TLD 后缀 .net 或 .ru, 如 windowafraid.net、mouthkind.ru 等。Matsnu 的 MAGD 是从嵌入恶意软件的两个单词列表中提取的单词的组合, 采用固定的 TLD 后缀 .com, 如 saladdoctortrainer.com。而 Cryptolocker 和 Locky 的分类准确率低于 0.5, 其中

Cryptolocker 被分类为 Tinba 的概率接近 0.16, 而被分类为 Murofet 的概率超过了 0.24; Locky 被分类成 Qakbot 的概率有 0.13 左右。

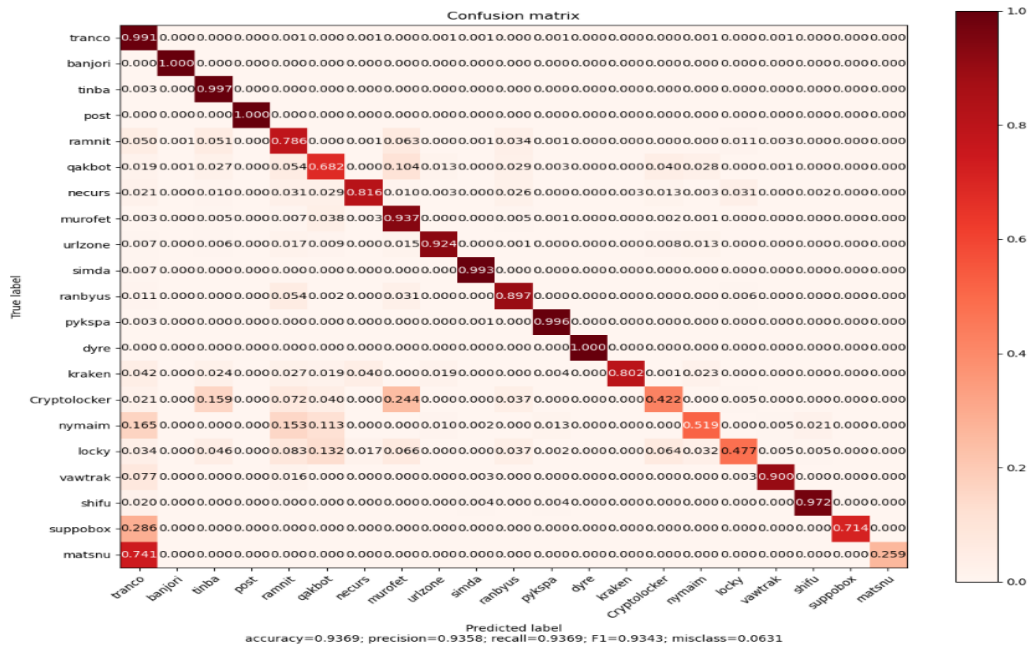


图 3.3 MAGD 多分类混淆矩阵

根据 DGArchive (<https://dgarchive.caad.fkie.fraunhofer.de/site/>) 记录的 DGA 特性可知: Tinba 是一种网络银行木马, 其正则表达式为 $[a-y]\{12\}\backslash\{2,7\}\$$, 即域名前缀长度固定 12 位, 均由字母 a 到 y 构成, 后缀是长度 2 到 7 的任意合法 TLD; Murofet 的正则表达式为 $[a-z]\{8,16\}\backslash\{com|net|org|info|biz\}\$$; Cryptolocker 是一种感染 Windows 平台的勒索软件, 其正则表达式为 $[a-y]\{12,15\}\backslash\{net|biz|org|com|ru|info|co.uk\}\$$ 。Cryptolocker 的可选 TLD 完全覆盖了 Murofet 可用的部分, 且 Murofet 的字符串长度范围完全覆盖了 Cryptolocker。在字符构成相近的情况下, 比起可选 TLD 更多、字符串长度固定的 Tinba, Cryptolocker 的 MAGD 与 Murofet 的更加接近。类似地, Qakbot 是于 2013 年被发现的银行木马程序, 其正则表达式为 $[a-z]\{8,25\}\backslash\{com|net|org|info|biz\}\$$; Locky 勒索软件于 2016 年初首次被发现, $[a-y]\{5,18\}\backslash\{be|biz|click|de|eu|fr|in|info|it|nl|org|pl|pm|pw|ru|su|tf|uk|us|work|xyz|yt\}\$$ 是它的正则表达式。在字符构成、字符串长度范围、可选 TLD 三个方面, Locky 都与 Qakbot 有一定的相似性。

综上, 集成模型对于基于单词的 Suppbob 和 Matsnu, 以及基于字符的 Cryptolocker 和 Locky 的分类准确率较低。在已有数据集的基础上, 可考虑适当增加这些家族的 MAGD 训练数据, 以优化集成模型分类效果。

3.3 本章小结

针对深度学习分类模型对域名信息利用率低,难以准确分类基于单词的 MAGD 等随机性较低的 DGA 域名的问题,本章提出了一种融合 PCNN 和结合注意力机制的 BiGRU 的 PCBGA-DGA 集成模型,该模型主要由预处理、字符嵌入、特征提取、输出分类结果四部分组成。该模型的特征提取层包括两大部分:一部分由 4 个卷积核大小不同的并行的 CNN 提取层组成,用于提取域名序列的 2 字符至 5 字符的局部特征;另一个部分由 BiGRU 结合自注意力机制组成,该提取层学习域名序列的双向时序依赖信息及字符权重信息。之后,特征提取层融合收集到的两种特征信息,送入全连接层以进行分类。通过对比实验及消融实验证明,相比其它模型,该集成模型精确度、召回率、F1 分数等多个评估指标上均有明显优势,对包括数量稀少且分类难度高的基于单词的 Suppobox 和 Matsnu 等多数 DGA 有最优的分类效果。最终,以模型的混淆矩阵为出发点,进一步分析分类效果较差的 DGA 家族的特性及其关系。

第四章 基于字符级替换的新型 DGA 域名生成及防御方法

基于深度学习的 DGA 域名检测方法, 具有低费、省时、自动表征、高效等优点, 但在图像、文本、音频等多个应用领域, 深度学习技术也带来了对抗攻击等巨大的安全问题^[57]。

在 DGA 恶意域名检测领域, 深度学习也已受到了来自对抗攻击的挑战, 有研究人员精心构建了使深度学习模型难以分类的新型 DGA。相比已知的 DGA, 深度学习技术更难检测新型 DGA, 因此新型 DGA 生成的恶意域名可用于增强僵尸网络的隐蔽性, 使攻击者与受控主机之间的 C&C 通信信道更难被轻易发现并封禁。且 DGA 算法仍在不断进化, 每次大规模的网络安全事件爆发, 总会有新型的 DGA 家族被发现。因此, 探索并预测新型 DGA 家族可能的生成方法, 以及提高深度学习技术检测这些新型 DGA 家族的性能十分重要。本章提出了一种基于字符级替换的新型 DGA 生成方法, 称为 CLR-DGA (Character-level replacement DGA), 它通过一定条件下的字符替换, 基于良性域名生成对抗攻击数据, 不需要了解 DGA 深度学习检测模型的细节即可取得比三种已知 DGA 及两种新型 DGA 更好的回避效果。本章还尝试使用对抗训练方法以防御这一新型 DGA, 以增强深度学习检测模型的鲁棒性。

4.1 对抗攻击

对抗攻击通过生成特殊数据, 使模型将特殊数据错误分类, 进而得到攻击者期望的检测结果, 是一种针对机器学习模型完整性的攻击。根据攻击者可能掌握的信息, 对抗攻击可分类为黑盒攻击 (Black-box attacks) 与白盒攻击 (White-box attacks)。在黑盒攻击中, 攻击者只能得知模型的输入和输出信息; 在白盒攻击中, 攻击者还可以获得模型的结构和参数等详细信息。根据攻击者干扰模型的时机, 对抗攻击可分类为投毒攻击 (Poisoning attacks) 与逃避攻击 (Evasion attacks)。投毒攻击指攻击者干扰机器学习模型的训练阶段, 通过攻击训练数据集或训练算法, 操纵机器学习模型的预测结果。逃避攻击指攻击者干扰机器学习模型的测试阶段, 通过精心设计难以被准确分类的数据回避机器学习模型的检测, 进而达成攻击目的。

在 DGA 恶意域名即 MAGD 检测领域, 一些研究者以良性域名或 MAGD 作为输入数据, 基于对抗攻击思想, 生成了新型 DGA 域名, 即对抗性域名

(Adversarial domain names)。相比主流的基于字符的 DGA 和基于单词的 DGA, 新型 DGA 在设计时多数采用黑盒逃避攻击的思路, 在降低模型分析成本的同时保留攻击的泛用性, 对机器学习尤其是深度学习方法具有很强的针对性。

4.2 CLR-DGA

CLR-DGA 是一种基于字符级替换 (Character-level replacement, CLR) 的新型 DGA, 可用于执行黑盒逃避攻击。CLR-DGA 通过替换良性域名字符串中的少量字符, 生成对抗性域名, 进而干扰深度学习分类器的检测结果。

该方法的灵感来自于“误植”(Typosquatting), 这是网络钓鱼和社会工程学领域的一种常用的技术^[58], 也是早期不依赖 DGA 而生成恶意域名的一种方法。

“误植”基于一个良性域名, 引入一些人类用户不太可能注意到的排版错误 (例如, 将 <http://www.google.com> 改为 <http://www.googlee.com>)。

本文将域名分为左侧域名和右侧域名两个部分。以 cnwomen.com.cn、baidu.com 等良性域名为例, 由于数据集中多数顶级域名和二级域名长度有限且搭配固定, 字符替换空间小且替换容易丢失域名真实性, 故仅替换从左往右第一个“.”的左侧域名 (也可称为“最低级域名”) 中的字符, 而其余部分即右侧域名保持不变。CLR-DGA 以良性域名字符串作为输入数据, 由左向右遍历字符串寻找第一个“.”, 将良性域名字符串分为左侧域名和右侧域名两部分。之后右侧域名不变, 左侧域名根据长度的不同执行不同的替换策略: (1) 左侧域名长度小于 4 时, 放弃对当前良性域名的替换操作; (2) 左侧域名长度是 4 或 5 时, 按照一定的约束条件 (如“-”不可出现在左侧域名的第一个字符和最后一个字符的位置等), 从左侧域名中随机选择两个字符进行替换; (3) 左侧域名长度大于等于 6 时, 按照与 (2) 相同的约束条件, 从左侧域名中随机选择三个字符进行替换。

(4) 若左侧域名执行了字符替换操作, 最后将左侧域名与右侧域名拼接, 即可得到一个 CLR-DGA 生成的域名。

由于对抗攻击有“细微干扰”的特点, 对于任意长度的最低级域名, 其对抗性域名中发生变化的字符数量不应超过最低级域名总长度的一半^[38]。同样是基于良性域名替换字符的方法, Charbot^[37]仅替换最低级域名的任意两个字符, 既不限最低级域名的总长度, 也未考虑最低级域名的第一个字符与最后一个字符不能为“-”的限制, 并且未考虑最低级域名仅有 2 到 3 个字符时仍替换任意 2 个字符导致信息修改过多的问题。而 CLR-DGA 考虑到了这三点, 使得生成的域名的字符特征更加接近良性域名。CLR-DGA 的算法步骤如图 4.1 所示。

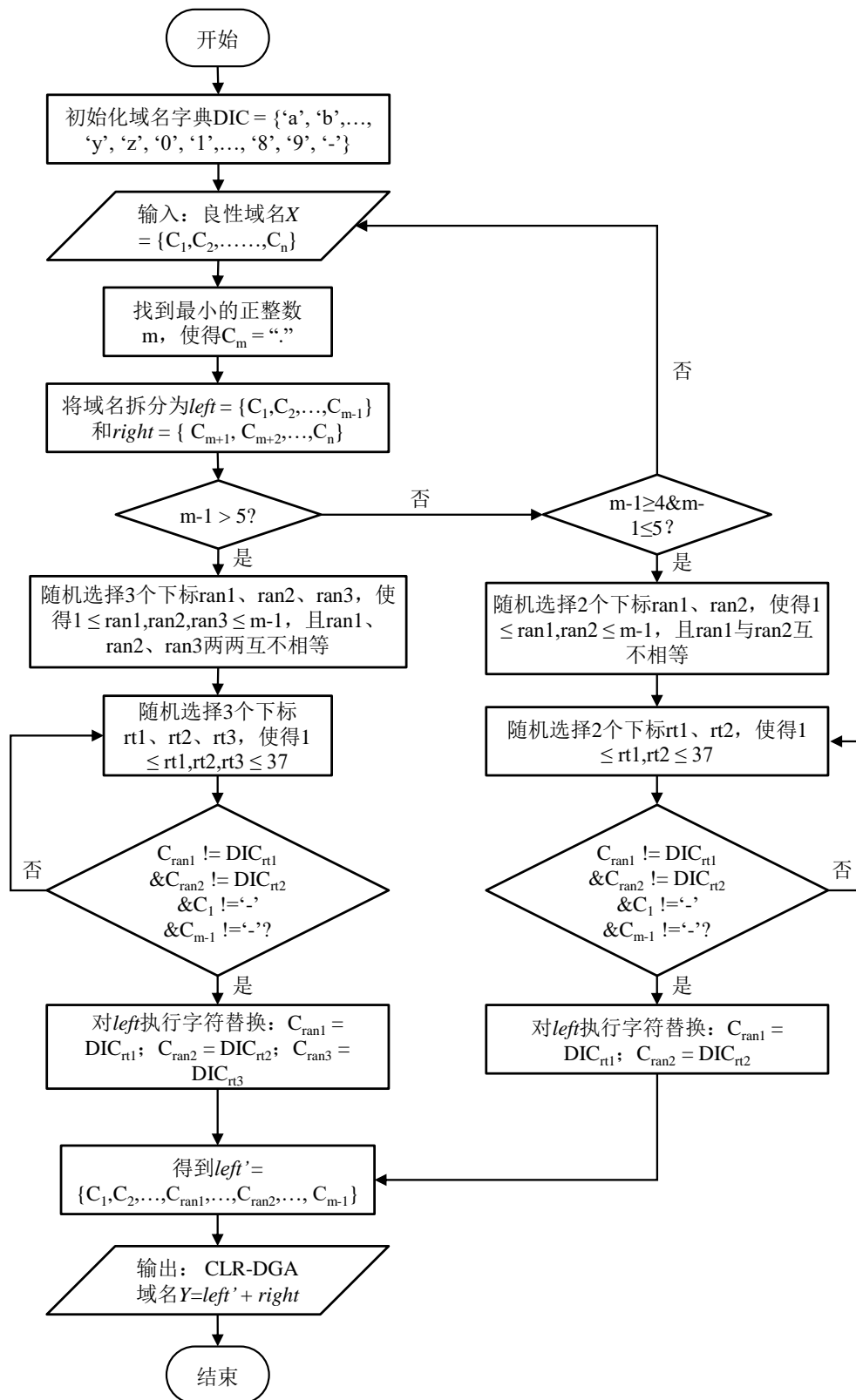


图 4.1 CLR-DGA 算法流程图

对抗攻击总是伴随着对抗成本函数 $\text{cost}(x, x)$, 该函数描述了将原始样本 x 扰

动为可用于攻击的样本 x 所需的成本。在典型的对抗性生成场景中，梯度被用来描述从原始样本到对抗样本的变化。但是，在 DGA 恶意域名的黑盒攻击场景中，计算梯度是不可行的，因为模型参数是不可知的。对文本进行对抗性修改的一个可行定义是文本 x 和文本 x 之间的编辑距离^[59]被定义为将 x 更改为 x 所需的最小编辑操作，编辑操作包括字符的增删改查。CLR-DGA 是基于字符替换的方法，因此可使用编辑距离定义成本函数，该函数定义如公式(4.1)所示：

$$\text{cost}(x, x) = \begin{cases} d_L(x, x), & x \text{ 是未注册域名} \\ \infty, & x \text{ 是已注册域名} \end{cases} \quad (4.1)$$

其中， d_L 表示 Levenshtein 距离或编辑距离。成本函数 $\text{cost}(x, x)$ 随着将 x 转换为 x 所需的编辑次数的增加而增加。每次编辑都可能使域名随机性增强，进而使得域名被 DGA 分类器检测到的概率有可能增大。而一个已经注册的域名无法被攻击者使用，生成这样的域名时，认为成本是无穷大。

CLR-DGA 旨在对抗成本和域名的可替换空间中取得平衡。设 l 为最低级域名长度， k 为替换字符数， d 为域名合法字符数 38。对 1 个良性域名，令 $l=10$ ，当替换字符数 $k=3$ 时，CLR-DGA 域名可能的组合数最多为 $C(l, k)(d-1)^k = 6,078,360$ 。而 $k=2$ 时， $C(l, k)(d-1)^k = 61,605$ 。相比于原理相近的 Charbot，CLR-DGA 可以选择的对抗性域名数量可增加数十倍，而成本函数仅提升了 50%。

从实用性的角度考虑，如果 DGA 可以生成大量尚未注册的域名，则 DGA 是成功的。Tranco 排名前 12 万的域名中，最低级域名的长度分布如图 4.2 所示。

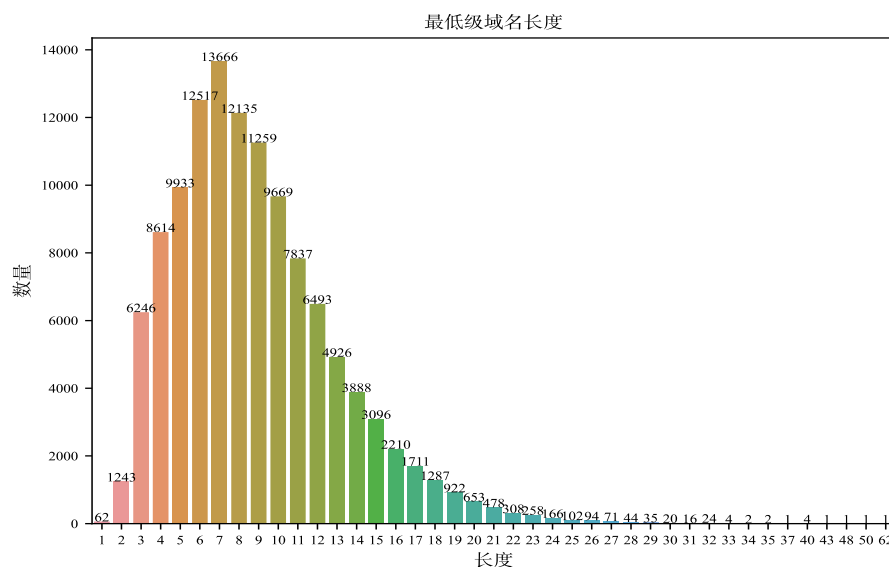


图 4.2 Tranco 前 12 万域名中，最低级域名长度分布

由图 4.2 可知,最低级域名长度小于等于 3 的短域名占比为 $(62+1243+6246)/120000 \approx 6\%$ 。长度为 3 的短域名最多有 $37^3=50653$ 个。相较于数以亿计的已注册域名^[60],3 个字符及以下的可选短域名数量少,在所有域名中比例小,且由于其商业价值,短域名大多已注册^[34, 61]。综上,不考虑生成这部分良性域名的对抗性域名。

一些由 CLR-DGA 生成的对抗性域名如表 4.1 所示。从字符串构成来看,对于由单词构成的良性域名,随机替换字符的方法可能会丢失一小部分单词语义,如 the-american-interest 中 interest 变为 inkerjxt, tmall 变为 tmaqo;但也有可能在不减少单词数量的情况下,使语义发生变化,如 ahhr 变为 aha0, freenem 变为 treen53, asian-singles 变为 asian-sin1wea。而对于随机性强的 xn--80afcdbalict6afooklqi5o,变化前后最低级域名部分均无明显规律。

表 4.1 CLR-DGA 域名示例

Tranco	↔	CLR-DGA
freenem.com	↔	treen53.com
secretbenefits.com	↔	secretbeneg5tj.com
boss-anime.com	↔	boss-znifc.com
beeinspiredclothing.com	↔	beeinspire4cloth8zg.com
asian-singles.net	↔	asian-sin1wea.net
the-american-interest.com	↔	the-american-inkerjxt.com
1398.org	↔	x338.org
ahhr.com.cn	↔	aha0.com.cn
tmall.com	↔	tmaqo.com
unwto.org	↔	duwto.org
viagrawithoutdoctorspres.com	↔	viagrwithoutdoctorsp8er.com
xn--80afcdbalict6afooklqi5o.xn--plai	↔	xn--86afsdcalict6afooklqi5o.xn--plai

对 12 万 Tranco 域名及生成的所有 CLR-DGA 域名的最低级域名字符频率分析如图 4.3 所示。CLR-DGA 随机替换的部分,域名中每个合法字符出现的概率是均等的。而域名随机性越强,意味着其中各字符的频率会越接近,直到所有字符频率相同。因此,在保留良性域名字符分布规律的基础上,与原本的 Tranco 域名相比,CLR-DGA 域名各字符的频率与 $1/37 \approx 0.027$ 更加接近,图 4.3 中的虚线对应字符频率 0.027。

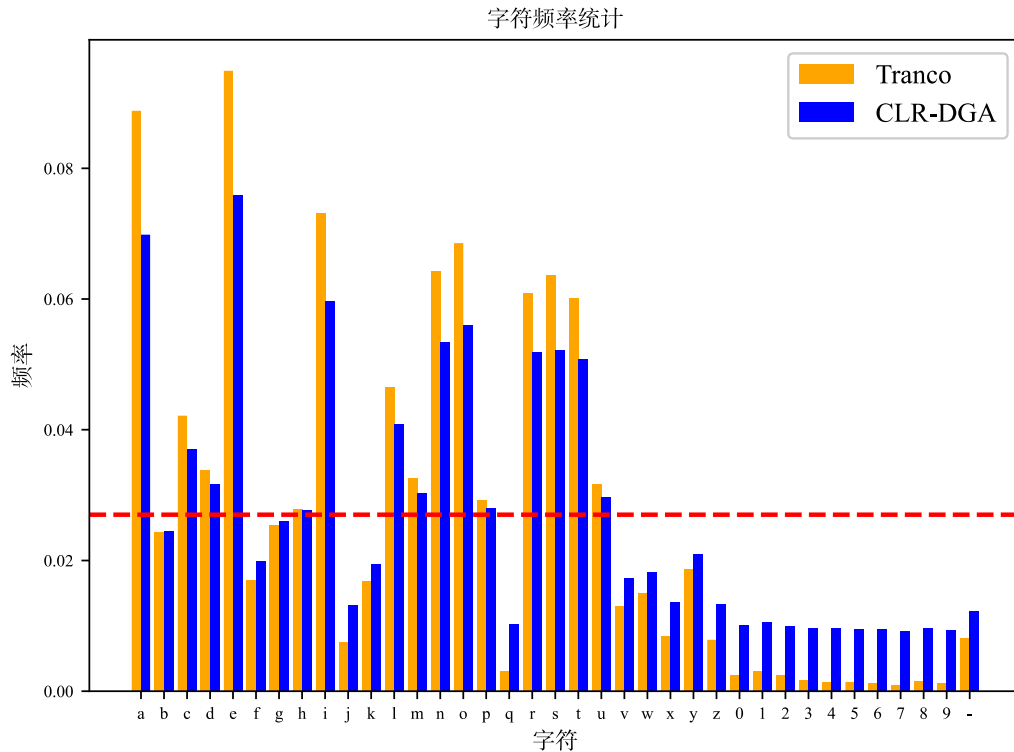


图 4.3 Tranco 和 CLR-DGA 的字符频率对比

4.3 实验与分析

4.3.1 实验环境与分类模型

实验环境由 Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, 15 GB 内存, GTX 1080 Ti 11GB GPU 和 Ubuntu 18.04 操作系统组成, 代码基于 Python 3.6 和 Tensorflow 1.14 编写。

深度学习分类模型采用第三章的 PCNN, LSTM-Att, Bilbo, CL, PCBGA-DGA 共 5 种模型。各模型最后的全连接层采用 2 个神经元对应良性和 DGA 恶意域名 2 种类别, 以对域名进行二分类, 测试深度学习模型对测试集出现的未知域名的检测效果。全连接层采用 Softmax 激活函数计算并输出分类结果。各模型其余参数设置与第三章相同。

4.3.2 数据集

良性域名选自 Tranco, 按照顺序选取其中排名前 12 万的域名。

DGA 域名选自 DGArchive^[22]。此外, 本文使用 Anderson 等人公开的源代码

重新实现了 DeepDGA, 并根据原理复现了 Charbot。包括本文提出的 CLR-DGA, 共有 16 个 DGA 家族, 其中“长度”与“字符”均统计最低级域名。根据生成方法, 它们可以分为五大类型:

基于算术: 随机组合字母和数字得到域名。

基于哈希: 通过哈希值生成域名。

基于单词: 拼接英文单词生成域名。

深度学习: 通过 GAN 网络等深度学习方法生成的对抗性域名。

字符替换: 基于字符替换方法生成的对抗性域名。

基于以上数据, 构成以下两个训练数据集和 6 个测试数据集, 如表 4.2 所示。模型训练期间, 训练集中 80% 的数据用于训练模型, 其余 20% 的数据作为验证集。

表 4.2 DGA 相关信息

DGA 家族	类型	长度	字符	目的	数量
Banjori	基于算术	7~30	[a-z]	训练集 1、2	10,000
Conficker	基于算术	5~11	[a-z]	训练集 1、2	10,000
Corebot	基于算术	12~23	[a-y0-8]	训练集 1、2	10,000
Dyre	基于哈希	34	[a-z0-9]	训练集 1、2	10,000
Matsnu	基于单词	12~24	[a-z]	训练集 1、2	10,000
Necurs	基于算术	7~26	[a-y]	训练集 1、2	10,000
Pykspa	基于算术	6~15	[a-z]	训练集 1、2	10,000
Qakbot	基于算术	8~25	[a-z]	训练集 1、2	10,000
Rovnix	基于算术	18	[a-z1-8]	训练集 1、2	10,000
Simda	基于算术	5~11	[a-z]	训练集 1、2	10,000
CLR-DGA1	字符替换	4~50	[a-z0-9-]	训练集 2	10,000
Bamital	基于哈希	32	[0-9a-f]	测试集 1	10,000
Ramnit	基于算术	8~19	[a-y]	测试集 2	10,000
Suppobox	基于单词	7~30	[a-z]	测试集 3	10,000
DeepDGA	深度学习	6~42	[a-z0-9-]	测试集 4	10,000
Charbot	字符替换	7~39	[a-z0-9-]	测试集 5	10,000
CLR-DGA2	字符替换	4~37	[a-z0-9-]	测试集 6	10,000

训练数据集 1: 10 万良性域名, 10 万 DGA 域名。良性域名来自 Tranco 排名前 10 万的域名, DGA 恶意域名来自 Banjori、Conficker、Corebot、Dyre、Matsnu、

Necurs、Pykspa、Qakbot、Rovnix、Simda 这 10 个 DGA 家族,每个家族从 DGArchive 取 1 万个域名。

训练数据集 2: 11 万良性域名, 11 万 DGA 域名, 用于 4.5 节的对抗训练。与训练数据集 1 相比, 增加了 Tranco 排名 10 万~11 万良性域名, 及 CLR-DGA 域名 1 万个。用于训练的这 1 万个 CLR-DGA 域名称为 CLR-DGA1。

测试数据集: Tranco 排名 11 万~12 万良性域名, 以及等量 DGA 家族域名。该数据集可用于评估深度学习分类器对 6 种未知 DGA 家族 (Bamital、Ramnit、Suppobox、DeepDGA、Charbot 和 CLR-DGA) 的检测效果。

虽然 Charbot 和 CLR-DGA 原理相似, 但由于生成方法有一定的随机性, 实际生成相同数量的 DGA 域名所用的良性域名集合不同。因此确保这两类域名构成的测试集的字符串长度、字符构成等分布相近即可。

4.3.3 评估指标

在第三章采用的精确度、召回率、F1 分数的基础上, 本章加入准确率 (Accuracy)。准确率表示模型分类正确的样本比例, 如公式(4.2)所示。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.2)$$

4.3.4 CLR-DGA 有效性验证

通过训练集 1 训练后的 5 个深度学习分类器, 对 6 个测试集的分类效果如表 4.3 所示。

横向分析表 4.3 中各分类模型对 CLR-DGA 及其它测试数据集的分类效果可知, PCNN、LSTM-Att、Bilbo、PCBGA-DGA 对 CLR-DGA 的分类效果在准确率、召回率和 F1 分数上表现最差; 对 CL 而言, 分类 Suppobox 时所有指标最低, 分类 CLR-DGA 时准确率、召回率和 F1 分数第二低。然而, CL 模型分类 Suppobox 的准确率不足 0.5, F1 分数仅为 0.22 左右, 远低于其余 4 个模型从 0.6193 到 0.7929 的准确率和从 0.5067 到 0.7853 的 F1 分数。Suppobox 完全由单词构成, 在不考虑字符权重时, 其顺序时序信息比其它 5 种 DGA 域名更接近良性域名; 但深度学习模型更全面地学习特征时, CLR-DGA 域名蕴含的特征信息比起其它 5 种 DGA 域名更接近良性域名, 使得分类器更难以准确分类。

纵向对比表 4.3 中含有 CLR-DGA 的测试数据集被不同的分类模型分类的效果可知, 检测 CLR-DGA 时, LSTM-Att 和 PCBGA-DGA 的各指标均低于其它 3

个分类器, 即 CLR-DGA 欺骗这两个分类器的效果更好。而表 3.3 中 LSTM-Att 和 PCBGA-DGA 对已知 DGA 的分类效果优于其它 3 个分类器。综上, 深度学习分类器学习常规 DGA 域名特征的效果越好, 就越有可能被 CLR-DGA 欺骗。

表 4.3 各模型分类 DGA 的准确度, 精确率, 召回率和 F1 分数

分类模型	评估指标	Bamital	Ramnit	Suppobox	DeepDGA	CharBot	CLR-DGA
PCNN	Accuracy	0.8693	0.9150	0.6795	0.6470	0.6060	0.5624
	Precision	0.7928	0.9391	0.7536	0.7819	0.6687	0.7455
	Recall	1.0000	0.8874	0.5334	0.4076	0.2851	0.1895
	F1-score	0.8844	0.9125	0.6247	0.5359	0.3998	0.3022
LSTM-Att	Accuracy	0.9632	0.9005	0.6193	0.5990	0.6540	0.5376
	Precision	0.9314	0.9332	0.7182	0.7308	0.7909	0.6813
	Recall	1.0000	0.8628	0.3929	0.3136	0.4187	0.1415
	F1-score	0.9645	0.8966	0.5079	0.4389	0.5475	0.2343
Bilbo	Accuracy	0.9047	0.9002	0.6268	0.5820	0.5630	0.5555
	Precision	0.8399	0.9625	0.7469	0.8012	0.6844	0.8014
	Recall	1.0000	0.8327	0.3834	0.2181	0.2338	0.1477
	F1-score	0.9130	0.8929	0.5067	0.3429	0.3485	0.2494
CL	Accuracy	0.9462	0.9028	0.4956	0.6372	0.5978	0.5619
	Precision	0.9028	0.9381	0.4858	0.7814	0.7038	0.7505
	Recall	1.0000	0.8624	0.1491	0.3811	0.3379	0.1856
	F1-score	0.9489	0.8987	0.2282	0.5123	0.4566	0.2976
PCBGA-DGA	Accuracy	0.9150	0.9137	0.7929	0.6108	0.5790	0.5378
	Precision	0.8546	0.9601	0.8151	0.7607	0.7244	0.7442
	Recall	1.0000	0.8633	0.7577	0.3233	0.2552	0.1152
	F1-score	0.9216	0.9091	0.7853	0.4538	0.3774	0.1995

在字符串文本中, Unigram (单字) 和 Bigram (双字) 包含了重要的特征。在域名字符串中, Unigram 指单个域名可用字符, 是组成域名字符串最基本的单位; Bigram 指任意两个域名可用字符的组合, 蕴含英文短语缩写和音节等特征^[34]。最低级域名中不存在“.”, 故此时域名合法字符可构成的 Bigram 最多有 $37 \times 37 = 1369$ 种。测试集采用的 7 类域名, 其最低级域名的 Bigram 分布如下: Tranco 是 1197 种, CLR-DGA 是 1369 种, Charbot 是 1228 种, Bamital 是 256 种, Ramnit 是 625 种, Suppobox 是 406 种, Deepdga 是 857 种。Charbot 和 CLR-DGA 的

Bigram 分布包含超过 Tranco 良性域名和其它测试用 DGA 的信息量, 却仍然难以检测。Bigram 分布说明 Charbot 和 CLR-DGA 虽然增强了随机性, 增加了冗余信息, 却仍比另外 4 种 DGA 有更加接近良性域名的特征。

从时间成本考虑, 基于 Tranco 排名前 12 万域名中最低级域名长度大于等于 4 的部分, 生成 112449 个 CLR-DGA 域名耗时 3.6 秒; 得到相同数量的 Charbot 域名耗时 3.56 秒。而 DeepDGA 模型预训练至少需要 14 小时, 才可生成域名。因此, 从生成 MAGD 所需时间考虑, CLR-DGA 是一种实用的对抗性域名生成方法。

从核心代码所需空间考虑, 实现 Charbot 需要 17983 字节的 Python 代码; 实现 CLR-DGA 需要 19020 字节的 Python 代码; 而 DeepDGA 算法需要在恶意软件中嵌入一个经过训练的机器学习模型, 该模型至少占用 6539192 字节。因此, 与 DeepDGA 相比, Charbot 和 CLR-DGA 更适于作为恶意软件内置的新型 DGA。

4.3.5 CLR-DGA 防御

CLR-DGA 类似于基于单词的 DGA, 两者都有一个字符串列表作为 DGA 生成的种子。基于单词的 DGA 将英文单词列表中若干个单词组合生成 DGA 域名, 而 CLR-DGA 在良性域名的基础上进行字符的替换。在这两种情况下, 生成的域名都表现出非常接近自然语言的属性, 这使得它们难以与良性域名区分开来。

为防御 CLR-DGA, 理论上最简单的防御方法是基于其生成原理, 获取相关域名; 并结合黑名单思想, 将其与完整的 Tranco 列表进行比较。如果该域名等于 Tranco 列表中替换若干个字符后生成的域名, 则该域名将被标记为恶意的域名。然而, Tranco 数据集包含数百万个样本。单个长度为 10 的最低级域名仅是替换 3 个字符就有 6 百万种可能, 而最低级域名长度主要分布在 4 到 22 之间, 每个长度都有大量的可替换域名。替换字符的数量可由攻击者控制。并且, 完整的 Tranco 有 750 万以上的良性域名; 攻击者还可以选择 Alexa 等其它良性域名列表。综上, 攻击者可以控制待替换的最低级域名长度、替换的字符数量、采用的良性域名数据集等要素, 使得该方法无效。

在深度学习领域, 对抗训练是有效防御对抗样本的方法。该方法用对抗样本扩充训练集, 对模型进行再训练。同理, 使用包含 1 万个 CLR-DGA 对抗性域名的训练集 2 对本章采用的 5 个深度学习模型进行对抗训练后, 各模型对 6 种 DGA 的检测效果如表 4.4 所示。表中↑表示对抗训练后某指标数值上升, ↓对抗训练后某指标数值下降。单个箭头表示数值变化幅度在 0 到 0.05 之间, 两个箭头表示数值变化幅度在 0.05 到 0.1 之间, 三个箭头表示数值变化幅度大于等于 0.1。

由表 4.4 可知, 经过对抗训练后, 5 个深度学习二分类模型对 CLR-DGA 的

所有指标均有 0.1 以上的涨幅,其中检测准确率由 0.5624, 0.5376, 0.5555, 0.5619, 0.5378 显著提升至 0.7200, 0.7645, 0.7478, 0.7400, 0.7779; F1 分数由 0.3022, 0.2343, 0.2494, 0.2976, 0.1995 显著提升至 0.6274, 0.7240, 0.6778, 0.6744, 0.7374。对抗训练后各模型检测 CLR-DGA 之外的 5 种 DGA 的各指标均有不同程度的波动: PCNN 模型对 Bamital 和 CharBot 的各指标均有提高,而对另外三种的检测指标总体呈下降趋势。LSTM-Att 模型对 Ramnit、DeepDGA 的各指标均有提高,而对其余 DGA 家族呈下降趋势。Bilbo 模型对 Suppobox 的各指标总体呈下降趋势,而对其余 DGA 家族呈上升趋势。CL 模型与 PCBGA-DGA 模型,虽然对 Bamital 家族的各指标均有下降,但对其余 4 种 DGA 的各指标总体呈上升趋势,尤其是召回率有至少超过 0.05 的提升。

表 4.4 对抗训练的效果

分类模型	评估指标	Bamital	Ramnit	Suppobox	DeepDGA	CharBot	CLR-DGA
PCNN	Accuracy	0.9294(↑↑)	0.9059(↓)	0.6323(↓)	0.6099(↓)	0.6331(↑)	0.7200(↑↑↑)
	Precision	0.8762(↑↑)	0.9643(↑)	0.7589(↑)	0.7494(↓)	0.6959(↑)	0.9379(↑↑↑)
	Recall	1.0000	0.8429(↓)	0.3878(↓↓↓)	0.3304(↓↓)	0.4275(↑↑↑)	0.4713(↑↑↑)
	F1-score	0.9340(↑)	0.8995(↓)	0.5133(↓↓↓)	0.4586(↓↓)	0.5296(↑↑↑)	0.6274(↑↑↑)
LSTM-Att	Accuracy	0.8944(↓↓)	0.9101(↑)	0.5814(↓)	0.7015(↑↑↑)	0.6165(↓)	0.7645(↑↑↑)
	Precision	0.8257(↓↓↓)	0.9110(↓)	0.6393(↓↓)	0.7476(↑)	0.7178(↓↓)	0.8743(↑↑↑)
	Recall	1.0000	0.9090(↑)	0.3736(↓)	0.6083(↑↑↑)	0.3839(↓)	0.6178(↑↑↑)
	F1-score	0.9045(↓↓)	0.9100(↑)	0.4716(↓)	0.6708(↑↑↑)	0.5003(↓)	0.7240(↑↑↑)
Bilbo	Accuracy	0.9124(↑)	0.9210(↑)	0.6099(↓)	0.6227(↑)	0.5800(↑)	0.7478(↑↑↑)
	Precision	0.8510(↑)	0.9615(↓)	0.6955(↓↓)	0.7458(↓↓)	0.7395(↑↑)	0.9380(↑↑↑)
	Recall	1.0000	0.8770(↑)	0.3911(↑)	0.3721(↑↑↑)	0.2472(↑)	0.5306(↑↑↑)
	F1-score	0.9195(↑)	0.9173(↑)	0.5007(↓)	0.4965(↑↑↑)	0.3705(↑)	0.6778(↑↑↑)
CL	Accuracy	0.9141(↓)	0.9273(↑)	0.4849(↓)	0.7520(↑↑↑)	0.6142(↑)	0.7400(↑↑↑)
	Precision	0.8534(↓)	0.9397(↑)	0.4616(↓)	0.7830(↑)	0.7491(↑)	0.9019(↑↑↑)
	Recall	1.0000	0.9132(↑↑)	0.1816(↑)	0.6971(↑↑↑)	0.3434(↑)	0.5386(↑↑↑)
	F1-score	0.9209(↓)	0.9263(↑)	0.2607(↑)	0.7376(↑↑↑)	0.4709(↑)	0.6744(↑↑↑)
PCBGA-DGA	Accuracy	0.8931(↓)	0.9387(↑)	0.7913(↓)	0.7415(↑↑↑)	0.6622(↑↑)	0.7779(↑↑↑)
	Precision	0.8238(↓)	0.9328(↓)	0.7506(↓↓)	0.7433(↓)	0.7824(↑↑)	0.9016(↑↑↑)
	Recall	1.0000	0.9456(↑↑)	0.8724(↑↑↑)	0.7378(↑↑↑)	0.4494(↑↑↑)	0.6238(↑↑↑)
	F1-score	0.9034(↓)	0.9392(↑)	0.8069(↑)	0.7405(↑↑↑)	0.5709(↑↑↑)	0.7374(↑↑↑)

以上结果表明,基于 CLR-DGA 的对抗训练可以有针对性地提升深度学习模型对 CLR-DGA 域名的分类效果,进而增强模型应对 CLR-DGA 的鲁棒性。其次,对抗训练后各模型对其它类别的 DGA 域名分类效果变化不稳定,但对于结构较为复杂、能够学习域名序列多种特征信息的 Bilbo、CL、PCBGA-DGA 这三种模型,基于 CLR-DGA 的对抗训练有助于它们对多种类型的 DGA 家族的分类效果。并且对抗训练后的模型对 Bamital、Ramnit 等常见 DGA 的检测效果仍明显优于 CLR-DGA,即对抗训练后的模型仍认为 CLR-DGA 比 Bamital、Ramnit 等更加接近良性域名,这侧面说明了 CLR-DGA 作为新型 DGA 具备难以被检测的特性,且深度学习模型对 CLR-DGA 等新型 DGA 的检测效果仍有提升空间。

4.4 本章小结

探索并预测新型 DGA 家族可能的生成方法,以及提高深度学习技术检测这些新型 DGA 家族的性能十分重要。本章首先介绍了对抗攻击的相关知识,之后提出了名为 CLR-DGA 的一种基于字符级替换的新型 DGA,并基于该 DGA 展开了对抗攻击和防御研究。在对抗攻击中,将 CLR-DGA 与 2 种基于字符的 DGA、1 种基于单词的 DGA,以及基于深度学习方法的 DeepDGA 和基于字符替换的 Charbot 两种新型 DGA 比较,观察 5 种深度学习分类器对以上 6 种 DGA 的检测效果。最后,针对本文提出的新型 DGA 进行了一定的防御研究,结果表明对抗训练方法有助于提升深度学习分类器应对 CLR-DGA 的鲁棒性。

第五章 总结与展望

5.1 本文总结

近年来发展迅猛的互联网充分展示了其两面性,在大幅便利人们生活的方方面面的同时,网络空间的安全隐患滋生了许多问题。作为有效的网络攻击平台之一,僵尸网络使个人和社会遭受了巨大损失,给网络安全造成了极大的威胁。而 DGA 域名生成技术的出现,使得僵尸网络更加强悍。DGA 域名检测技术有助于使网络安全防御者及时发现僵尸网络并阻断其通信,而结合深度学习技术可使 DGA 域名检测手段更加智能化。本文主要基于深度学习模型,对 DGA 域名的分类、生成和防御进行了研究。

(1) 在 DGA 域名分类检测方面,本文提出了一种基于深度学习的 DGA 域名分类模型 PCBGA-DGA。该模型利用并行的 CNN 和结合注意力机制的 BiGRU,分别提取域名序列的局部特征,和包含字符权重信息的上下文的时序依赖特征;之后融合这两种特征信息,使用深度学习模型对公开的域名数据集开展多分类实验。实验结果表明,相比传统的深度学习模型,该模型对 DGA 域名有更优的多分类效果,在分类基于单词列表生成的两种 DGA 域名时也有明显的优势。

(2) 在新型 DGA 的生成及防御方面,本文提出了 CLR-DGA,一种基于字符级替换的新型 DGA,通过基于良性域名的字符级替换生成难以检测的域名。之后本文基于该 DGA 展开了对抗攻击和防御研究。在对抗攻击实验中,将 CLR-DGA 与已知的基于算术的 DGA、基于哈希的 DGA、基于单词的 DGA 各一种,以及基于深度学习方法的 DeepDGA 和基于字符替换的 Charbot 两种新型 DGA 比较,观察 5 种深度学习分类器对以上 6 种 DGA 的检测效果。最后,针对本文提出的新型 DGA 进行了一定的防御研究,结果表明通过对抗训练方法可以在一定程度上增强深度学习分类器抵御来自 CLR-DGA 的攻击,进而提升分类器的鲁棒性。

5.2 未来展望

本文结合深度学习技术,开展 DGA 域名的分类与检测相关研究,取得了一定的成果,但依然存在不足之处,有待进一步研究完善。

(1) 已知的 DGA 家族种类繁多, DGArchive 数据集中已有近 100 种 DGA

家族，而本文提出的检测方法仅验证了对 20 种 DGA 家族的多分类效果，因此检测模型的泛化能力有待优化，之后的研究工作可考虑在数据量更加庞大且 DGA 家族更多的域名数据集上进行验证。

（2）新型 DGA 的生成方法不只有字符级替换，还包括结合 GAN 网络技术的 DeepDGA，基于机器学习特征工程的 Deception_DGA 等。为更好地探索并预测新型 DGA 家族可能的生成方法，进而分析新型 DGA 可能具备的特征，通过对抗训练等方式增强深度学习等 DGA 域名检测方法的鲁棒性，有必要继续探索新型 DGA 的生成策略。

（3）对抗训练虽然有助于在一定程度上增强深度学习分类模型的鲁棒性，但其更新模型的效率较低。有必要研究更有效的方法，以增强模型的鲁棒性。

参考文献

- [1] 中国互联网络信息中心. 第 50 次中国互联网络发展状况统计报告[R/OL]. (2022-08-31)[2022-12-15]. <http://www.cnnic.net.cn/n4/2022/0914/c88-10226.html>.
- [2] 中国信息安全测评中心. 2022 上半年网络安全漏洞态势观察[R/OL]. (2022-09-02)[2022-12-15]. http://www.itsec.gov.cn/zxxw/202209/t20220902_112723.html.
- [3] Hancock B. US and Europe cybercrime agreement problems[J]. Computers & Security, 2000, 19(4): 306-306.
- [4] Johnson M E. Managing information risk and the economics of security[M]. Boston, MA: Springer Boston MA, 2009: 1-16.
- [5] Sai Charan P V, Gireesh Kumar T, Mohan Anand P. Advance persistent threat detection using long short term memory (LSTM) neural networks[C]. International Conference on Emerging Technologies in Computer Engineering. Singapore: Springer, 2019: 45-54.
- [6] 国家互联网应急中心. 2021 年上半年中国互联网网络安全监测数据分析报告[R/OL]. (2021-07-31) [2022-12-15]. https://www.cert.org.cn/publish/main/46/2021/20210731090556980286517/20210731090556980286517_.html.
- [7] 崔丽娟, 马卫国, 赵巍, 等. 僵尸网络综述[J]. 信息安全研究, 2017, 3(07): 589-600.
- [8] Plohmann D, Yakdan K, Klatt M, et al. A comprehensive measurement study of domain generating malware[C]. 25th USENIX Security Symposium (USENIX Security 16). Berkeley: USENIX Association, 2016: 263-278.
- [9] Al-Nawasrah A, Almomani A A, Atawneh S, et al. A survey of fast flux botnet detection with fast flux cloud computing[J]. International Journal of Cloud Applications and Computing (IJCAC), 2020, 10(3): 17-53.
- [10] 王媛媛, 吴春江, 刘启和, 等. 恶意域名检测研究与应用综述[J]. 计算机应用与软件, 2019, 36(09): 310-316.
- [11] Stone-Gross B, Cova M, Cavallaro L, et al. Your botnet is my botnet: analysis of a botnet takeover[C]. Proceedings of the 16th ACM conference on Computer and communications security. New York: ACM, 2009: 635-647.
- [12] Bilge L, Sen S, Balzarotti D, et al. Exposure: A passive dns analysis service to detect and report malicious domains[J]. ACM Transactions on Information and System Security (TISSEC), 2014, 16(4): 1-28.

- [13]Peng C, Yun X, Zhang Y, et al. Malhunter: Performing a timely detection on malicious domains via a single DNS query[C]. International Conference on Information and Communications Security. Cham: Springer, 2018: 685-695.
- [14]Schüppen S, Teubert D, Herrmann P, et al. FANCI: Feature-based automated nxdomain classification and intelligence[C]. 27th USENIX Security Symposium (USENIX Security 18). Berkeley: USENIX Association, 2018: 1165-1181.
- [15]Thomas M, Mohaisen A. Kindred domains: detecting and clustering botnet domains using DNS traffic[C]. Proceedings of the 23rd International Conference on World Wide Web. New York: ACM, 2014: 707-712.
- [16]Wang T S, Lin H T, Cheng W T, et al. DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis[J]. Computers & Security, 2017, 64: 1-15.
- [17]Antonakakis M, Perdisci R, Nadji Y, et al. From throw-away traffic to bots: detecting the rise of DGA-based malware[C]. 21st USENIX Security Symposium (USENIX Security 12). Berkeley: USENIX Association, 2012: 491-506.
- [18]Chin T, Xiong K Q, Hu C B, et al. A machine learning framework for studying domain generation algorithm (DGA)-based malware[C]. International Conference on Security and Privacy in Communication Systems. Cham: Springer, 2018: 433-448.
- [19]李晓冬, 李育强, 宋元凤, 等. 新的基于融合向量的 DGA 域名检测方法[J]. 计算机应用研究, 2022, 39(06): 1834-1837.
- [20]Rahbarinia B, Perdisci R, Antonakakis M. Segugio: efficient behavior-based tracking of malware-control domains in large ISP networks[C]. 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Piscataway: IEEE, 2015: 403-414.
- [21]Sun X Q, Yang J H, Wang Z L, et al. HGDom: heterogeneous graph convolutional networks for malicious domain detection[C]. NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. Piscataway: IEEE, 2020: 1-9.
- [22]Li Z, Yuan F, Liu Y, et al. Heterogeneous graph attention network for malicious domain detection[C]. International Conference on Artificial Neural Networks. Cham: Springer, 2022: 506-518.
- [23]Woodbridge J, Anderson H S, Ahuja A, et al. Predicting domain generation algorithms with long short-term memory networks[EB/OL]. (2016-11-02)[2022-11-22]. <https://arxiv.org/abs/1611.00791>.
- [24]Tran D, Mac H, Tong V, et al. A LSTM based framework for handling multiclass imbalance in DGA botnet detection[J]. Neurocomputing, 2018, 275: 2401-2413.
- [25]Tuan T A, Long H V, Taniar D. On detecting and classifying DGA botnets and their families[J]. Computers & Security, 2022, 113: 102549.
- [26]Yu B, Pan J, Hu J M, et al. Character level based detection of DGA domain

- names[C]. 2018 International Joint Conference on Neural Networks (IJCNN). Piscataway: IEEE, 2018: 1-8.
- [27] Saxe J, Berlin K. EXpose: A character-level convolutional neural network with embeddings for detecting malicious urls, file paths and registry keys[EB/OL]. (2017-02-27)[2022-11-22]. <https://arxiv.org/abs/1702.08568>
- [28] 刘小洋, 刘加苗, 刘超, 等. 融合字符级滑动窗口和深度残差网络的僵尸网络 DGA 域名检测方法[J]. 电子学报, 2022, 50(01): 250-256.
- [29] Zhao K, Guo W, Qin F, et al. D3-SACNN: DGA domain detection with self-attention convolutional network[J]. IEEE Access, 2021, 10: 69250-69263.
- [30] 杜鹏, 丁世飞. 基于混合词向量深度学习模型的 DGA 域名检测方法[J]. 计算机研究与发展, 2020, 57(02): 433-446.
- [31] Namgung J, Son S, Moon Y S. Efficient deep learning models for DGA domain detection[J]. Security and Communication Networks, 2021, 14: 8887881.
- [32] Liu X Y, Liu J M. DGA botnet detection method based on capsule network and k-means routing[J]. Neural Computing and Applications, 2022, 34(11): 8803-8821.
- [33] Anderson H S, Woodbridge J, Filar B. DeepDGA: adversarially-tuned domain generation and detection[C]. Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. New York: ACM, 2016: 13-21.
- [34] Yun X C, Huang J, Wang Y P, et al. Khaos: an adversarial neural network DGA with high anti-detection ability[J]. IEEE transactions on information forensics and security, 2019, 15: 2225-2240.
- [35] Zheng Y, Yang C, Yang Y, et al. ShadowDGA: Toward Evading DGA Detectors with GANs[C]. 2021 International Conference on Computer Communications and Networks (ICCCN). Piscataway: IEEE, 2021: 1-8.
- [36] Spooren J, Preuveneers D, Desmet L, et al. Detection of algorithmically generated domain names used by botnets: a dual arms race[C]. Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. New York: ACM, 2019: 1916-1923.
- [37] Peck J, Nie C, Sivaguru R, et al. CharBot: A simple and effective method for evading DGA classifiers[J]. IEEE Access, 2019, 7: 91759-91771.
- [38] Sidi L, Nadler A, Shabtai A. MaskDGA: An evasion attack against DGA classifiers and adversarial defenses[J]. IEEE Access, 2020, 8: 161580-161592.
- [39] 方滨兴, 崔翔, 王威. 僵尸网络综述[J]. 计算机研究与发展, 2011, 48(08): 1315-1331.
- [40] Davidson R. The fight against malware as a service[J]. Network Security, 2021, 2021(8): 7-11.
- [41] 王垚, 胡铭曾, 李斌, 等. 域名系统安全研究综述[J]. 通信学报, 2007, 28(009): 91-103.

- [42]Mockapetris P. Domain names-implementation and specification[EB/OL]. (1987-11-01)[2022-11-22]. <http://www.rfc-editor.org/rfc/rfc1035.txt>.
- [43]Godaddy. The top 25 most expensive domain names[EB/OL]. (2022-07-14)[2022-11-22]. <https://www.godaddy.com/garage/the-top-20-most-expensive-domain-names/>.
- [44]Plohmann D, Yakdan K, Klatt M, et al. A comprehensive measurement study of domain generating malware[C]. 25th USENIX Security Symposium (USENIX Security 16). Berkeley: USENIX Association, 2016: 263-278.
- [45]Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77: 354-377.
- [46]LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
- [47]Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [48]Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB/OL]. (2014-06-03)[2022-11-22]. <https://arxiv.org/abs/1406.1078>
- [49]Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB/OL]. (2016-05-19)[2022-11-22]. <https://arxiv.org/abs/1409.0473>.
- [50]Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673-2681.
- [51]Zhang J, Liu F, Xu W, et al. Feature fusion text classification model combining CNN and BiGRU with multi-attention mechanism[J]. Future Internet, 2019, 11(11): 237.
- [52]Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in Neural Information Processing Systems 30. New York: Curran Associates Inc, 2017: 5998-6008.
- [53]Qiao Y, Zhang B, Zhang W, et al. DGA domain name classification method based on long short-term memory with attention mechanism[J]. Applied sciences, 2019, 9(20): 4205.
- [54]Pochat V L, Van Goethem T, Tajalizadehkhoob S, et al. Tranco: A research-oriented top sites ranking hardened against manipulation[EB/OL]. (2018-12-17)[2022-11-22]. <https://arxiv.org/abs/1806.01156>
- [55]Highnam K, Puzio D, Luo S, et al. Real-time detection of dictionary DGA network traffic using deep learning[J]. SN Computer Science, 2021, 2(2): 1-17.
- [56]张永斌, 常文欣, 孙连山, 等. 基于字典的域名生成算法生成域名的检测方法[J]. 计算机应用, 2021, 41(09): 2609-2614.
- [57]Xu H, Ma Y, Liu H C, et al. Adversarial attacks and defenses in images, graphs and

- text: A review[J]. International Journal of Automation and Computing, 2020, 17(2): 151-178.
- [58]Szurdi J, Kocso B, Cseh G, et al. The Long "Taile" of Typosquatting Domain Names[C]. 23rd USENIX Security Symposium (USENIX Security 14). Berkeley: USENIX Association, 2014: 191-206.
- [59]Levenshtein V I. Binary codes capable of correcting deletions, insertions, and reversals[J]. Soviet physics doklady. 1966, 10(8): 707-710.
- [60]VeriSign. The domain name industry brief: Q2 2022 data and analysis[EB/OL]. (2022-6-30)[2022-11-22]. https://www.verisign.com/en_US/domain-names/dnib/index.xhtml?section=executive-summary.
- [61]Dynadot. Short available domain names: how to find them[EB/OL]. (2021-8-30) [2022-11-22]. <https://www.dynadot.com/community/blog/short-available-domain-names.html>.