基于改进 GAN 的恶意域名数据增强

傅 伟 钱丽萍 朱晓慧
(北京建筑大学电气与信息工程学院 北京 100044)

摘 要 近年来以恶意域名为依托的网络攻击事件频发。针对主流检测方法识别 DGA(Domain Generation Algorithm)变体域名面临的训练数据受限和时效性不足问题,提出一种基于改进 WGAN 模型的伪 DGA 域名生成方法。将 skip-gram 和 WGAN 结合,通过 skip-gram 完成域名有效转换,WGAN 模型深度挖掘数据编码中包含的特征,学习并生成伪 DGA 域名。为验证模型生成数据的有效性,采用多种机器学习方法对生成的域名进行有效性评估。实验结果表明,基于此模型生成的数据具备原数据的特性,可以模拟真实域名用于扩充恶意域名数据集,缓解现有域名检测算法中缺乏 DGA 变体域名的问题。

关键词 恶意域名 数据增强 域名生成算法 字符嵌入 生成对抗网络 检测

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.03.049

MALICIOUS DOMAIN NAME DATA AUGMENTATION BASED ON IMPROVED GAN

Fu Wei Qian Liping Zhu Xiaohui
(*College of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China*)

Abstract In recent years, cyber-attacks based on malicious domain names occur frequently. Aiming at the problem of limited training data and timeliness of DGA variant domain names in the mainstream detection method, a pseudo-DGA domain name generation method based on improved WGAN model is proposed. It combined skip-gram and WGAN to complet the effective conversion of domain names through skip-gram, and the WGAN model deeply mined the features contained in the data encoding, learned and generated pseudo-DGA domain names. To verify the validity of the generated data, a variety of machine learning methods were used to evaluate the validity of the generated domain name. The experimental results show that the data generated by this model has the characteristics of the original data, which can simulate the real domain name to expand the malicious domain name data set, and alleviate the lack of DGA variant domain name in the existing domain name detection algorithm.

Keywords Malicious domain name Data augmentation Domain generation algorithm Skip-gram Generative adversarial networks Detection

0 引 言

互联网技术为人们的日常工作和生活提供了便利,但互联网攻击事件也层出不穷,包括僵尸网络、网页钓鱼、网络窃听、托管诈骗等,严重侵犯用户隐私和威胁用户财产安全。CNCERT/CC2018 年度报告中指出,77 373 个服务器 IP 地址被木马或僵尸程序控制,其中境内有 6 559 208 个主机 IP 地址被控制。

域名系统(Domain Name System,DNS)是互联网的重要组成部分,为用户提供易于记忆的域名和 IP 地址映射的服务[1]。因其应用的广泛性以及缺乏内置的安全检测机制,常被攻击者用于承载网络攻击。僵尸网络即是依托域名系统,将用户正常域名破坏性地解析到恶意服务器上,从而达到控制用户主机的目的[2]。早期的安全检测系统较容易发现此类恶意域名,会迅

收稿日期:2019-11-06。国家自然科学基金项目(61571144)。**傅伟**,硕士生,主研领域:网络安全。**钱丽萍**,副教授。**朱晓慧**,硕士生。

速阻断通信并将其列入黑名单。为对抗黑名单机制,当前攻击者普遍引入域名生成算法 DGA,动态地生成恶意域名,绕过安全系统的检测,同时也显著增强了恶意服务器的持久性和隐蔽性[3]。与此同时,寻找高效快速检测 DGA 域名的方法成为网络安全领域的研究热点之一。

我们可以将域名检测方法大致分为两类:基于域名特征提取和基于无显性特征提取。域名特征包括域名内容特征、域名字符统计特征和域名解析行为间关系特征等。除人工提取的特征外,深度学习框架通过训练可以提取域名隐性特征[4]。但两类检测方法都是基于现有恶意域名进行检测,对于不断更新的 DGA 算法产生的新域名检测时效性不强。DGA 新域名规避某些传统检测特征的特性,以及其数据量少、采集困难、获取周期滞后,给域名检测算法带来极大挑战,因此恶意域名数据集增强意义重大。

1　相关工作

目前学术界在恶意域名对抗方面主要集中在检测方法。在基于域名内容或字符特征的检测方面,Schiavoni 等提出了 Phoenix 机制,不仅可以根据字符串和 IP 地址特征区分是否为 DGA 域名,还可以挖掘隐藏在 DGA 后的 Botnet,实验采用 115 万恶意域名,检测准确率在 94.8% 左右[5]。Mowbray 等在域名解析过程中,通过分析不常见的二元字符串分布来识别恶意域名[6]。Yadav 等利用 DNS 探测法统计域名一元、二元字符分布来探索域名隐含的特性,检测的方法是计算域名的 K-L 距离、Jaccard 距离和编辑距离[7]。除了传统的字符特征统计,文献[8]将域名划分成单个单词,达到扩展功能集尺度的目的,进而提高恶意域名识别准确率。Truong 等发现 DGA 域名和合法域名存在不同的构成规则,据此提出从 DNS 流量中提取长度和期望值以区分两种域名,构建的 J48 分类器平均准确率达到 92.3%,假阳性率为 4.8%[9]。

在域名解析行为的上下文关系的检测方面,Wang 等[10]基于僵尸主机会在同一域中查询大量域名且多数域名查询失败这一事实,结合 Botnet 检测困难和隐蔽性高等特点,提出了 DBod 的检测方案。文献[11]通过分析 DNS NXDomain 流量,结合 DGA 域名使用周期短且具有相似查询方式的特点,从而对 DGA 域名进行识别检测。

在无特征提取的检测方面,Yu 等[12]提出一种 LSTM + CNN 模型的深度学习检测方法,LSTM 模型学习域名字符序列的同时不丢失长期依赖的信息,相比于基于字符特征搭建的随机森林框架,该模型检测效果突出。Anderson 等[13]利用生成对抗网络模型,结合自动编码器生成对抗样本,以期得到与恶意域名数据集类似的数据集,通过随机森林分类器验证了对抗样本的有效性。

综上,上述研究方法除文献[13]外,均未考虑实时检测新域名问题。本文所做的工作正是为其奠定基础,通过生成伪 DGA 域名,扩充恶意域名数据集,满足黑名单系统和检测方法的实效性需求。与文献[13]不同之处在于:1) 本文采用 skip-gram 模型对域名进行编码,使得域名字符间的特征关系较好地反映在词向量中;2) 域名生成模型采用改进的 WGAN(Wasserstein GAN);3) 对实验生成的域名采用多种常见的分类器进行分析评估,使结果更具有说服力。

2　模型结构

本文针对域名数据增强问题,提出基于 skip-gram 数据编码加 WGAN 数据对抗生成的深度学习模型框架。

深度学习的快速发展解决了许多复杂问题,却无法直接识别字符串信息,因此本文在保留域名携带信息的前提下对其进行编码是需要关注的难点之一。Zheng 等[14]在处理网络文本语料时,采用 skip-gram 模型学习文本间的语义相关性,对文本主题嵌入建模,该方法表现了较好的性能[14]。skip-gram 作为自然语言处理领域的重要模型,对文本数据处理具有普适性。模型通过训练将文本转化为词向量,其语义的空间距离代表文本间的相似度,故空间距离可以近似表示文本相似度。深度学习中度量空间距离的方法很多,其中,效果较好的有皮尔逊相关系数,文献[15]基于皮尔逊相关系数研究评估网络舆情。皮尔逊相关系数能够从数学角度衡量文本的相似度。

自生成对抗网络(Generative Adversarial Networks, GAN)问世以来,基于 GAN 及其变体模型的数据集扩充的研究越来越多[16]。许春冬等[17]基于 CGAN 网络实现文本语音增强的目的,增强后的语音质量更高。蒋鹏飞等[18]采用改进的 GAN 网络生成网络事件序列样本,实验结果表明,生成数据有效且具有多样性。GAN 在训练中存在诸多问题,而 GAN 的变体模型在文本序列生成的研究中表现出较好的效果。

本文的生成模型采用 WGAN,在模型前后分别添加 skip-gram 编码器和皮尔逊相关系数解码器,如图 1 所示。

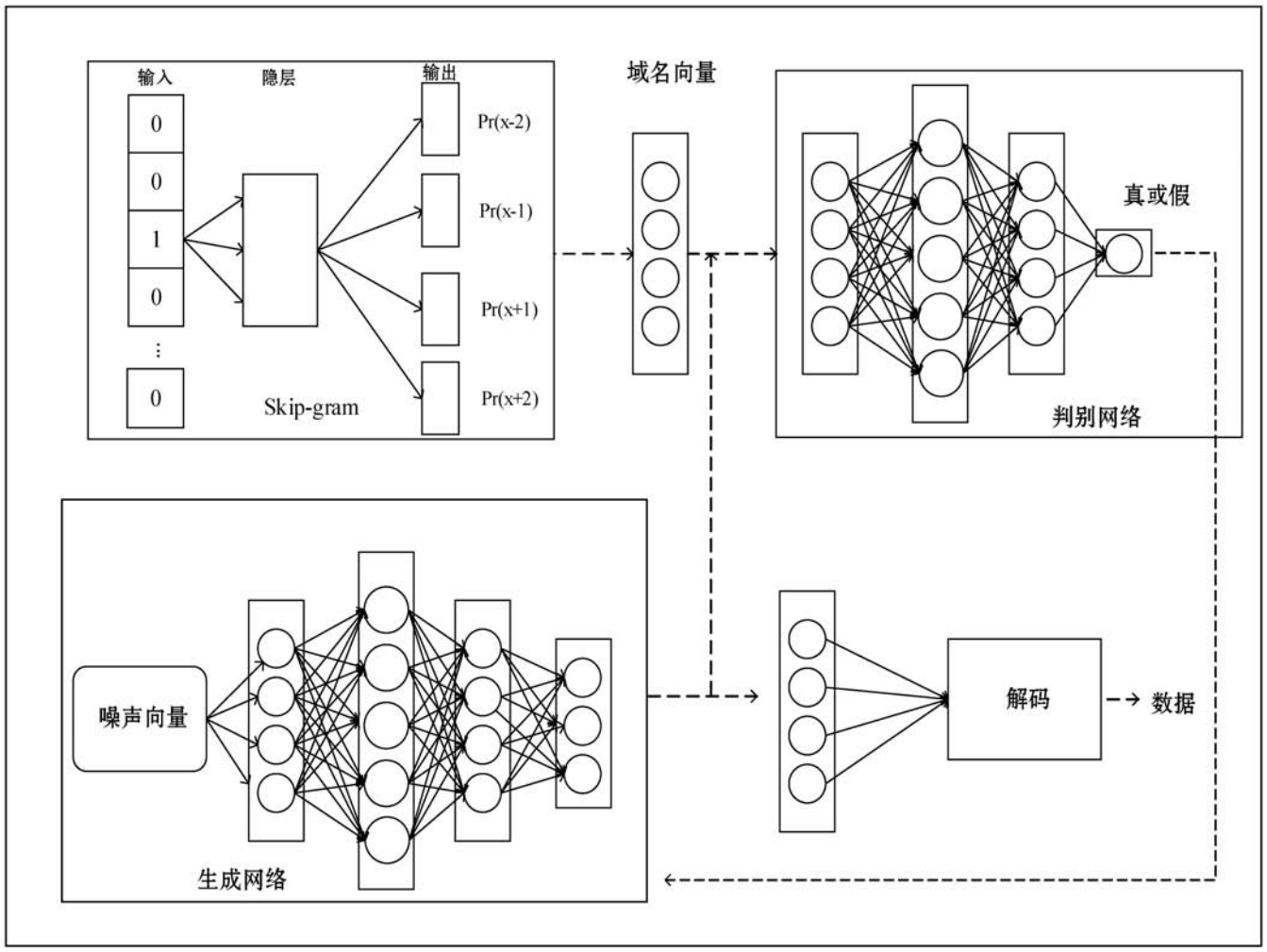

图 1 域名生成模型

2.1 域名编解码模型

2.1.1 skip-gram 模型

作为深度学习的文本输入,早期的词表示方法采用 one-hot,其缺点比较明显,词向量冗长且不能保留原文语义关系。skip-gram 模型作为 word embedding 的一种无监督学习方法,普适于各种文本,且能将其最小单元的特征映射到向量中[19-20]。模型主要思想是给定中心词预测上下文单词。相比于 CBOW 模型,该方法训练效率更高,速度更快,更精确地学习到域名特征,大大缩短后期 WGAN 学习域名的时间。

skip-gram 在自然语言处理领域研究的数据单元一般是英文句子的单词、中文句子的词语。域名本身与英文单词相似,所以本文处理的数据单元是最细粒度的字符。

模型分为输入层、隐层和输出层。其中输入层采用 one-hot 编码,隐层不使用激活函数,输出层采用 softmax 函数。模型使用负采样(Negative Sampling)技术。模型目标是最大化窗口内字符的概率,最小化未在窗口内字符的概率。

模型训练的过程如图 2 所示。阴影代表中心词 input-word,方框代表窗口。

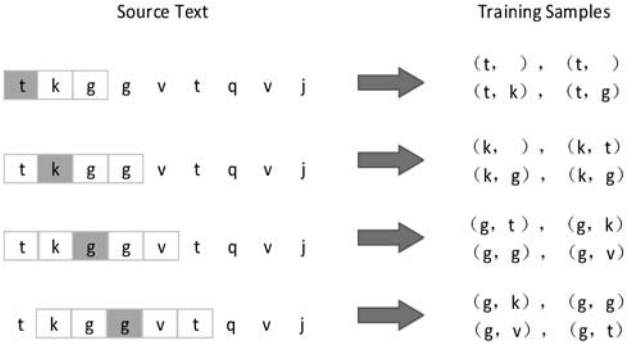

图 2 skip-gram 模型扫描域名示意图

首先从左到右选取一个字符当作中心词,设置窗口大小为 2,输出词会从窗口中选取,形成(中心词,输出词)训练样本,如(g,v)(g,k)。通过输入神经网络上述成对的样本进行训练。图 2 中(g,v)样本对出现两次,如果 g 作为输入数据,则输出 v 的概率比输出 q 的概率大,这只是针对单个单词的训练结果。模型训练的数据集是所有域名通过扫描得到的所有样本对。

模型损失函数如式(1)。

$$J(\theta) = \sum_{x,y\in Z} \frac{1}{1+\exp(-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})} - \sum_{x,y\in Z'} \frac{1}{1+\exp(-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})} \quad (1)$$

式中:$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 分别是字符 x 和 y 的向量表示,Z 是上述训练过程扫描得到的样本数据集,Z'是负采样数据集。

综上所述,利用滑动窗口得到的样本对,充分体现字符间的联系。与此同时,窗口的大小设置成 2,有效防止了字符距离过远而产生错误联系的问题。

2.1.2　字符编码

模型通过学习样本对进行不断训练,隐层的权重不断更新,最终会生成一个权重矩阵。隐层矩阵是[26×4]维,4 维是设置的每个单词映射的词向量维度,26 是字符总数,也就是字典的 size。模型训练结束后字典每个键都对应隐层的权重。

域名数据集每个字符都对应一个[1×4]的词向量,字符之间关系均体现在每个词向量中。所以将数据集中的每个字符对应映射即可,域名的编码结果样例如图 3 所示。

t	0.36640580	-0.04686514	-0.09865253	-0.92402280
k	0.27422702	0.27487215	-0.11145002	-0.91478080
g	0.74090856	-0.38805193	0.41818260	-0.35439170
g	0.74090856	-0.38805193	0.41818260	-0.35439170
v	0.15687898	0.70176256	0.55670780	0.41592620
t	0.36640580	-0.04686514	-0.09865253	-0.92402280
q	-0.30827108	0.18765156	-0.74186700	-0.56514513
v	0.15687898	0.70176256	0.55670780	0.41592620
j	0.21963617	0.31600773	-0.71061397	-0.58900490

图 3　编码结果样例

2.1.3　字符解码

由于本文域名字符级编码采用的是 skip-gram,解码时会涉及空间距离,采用皮尔逊相关系数度量字符间的相似性。该方法相比于欧氏距离,在数据不规范的时候也能给出较好的结果[21]。具体计算公式如式(2)。

$$\rho_{s,t} = \frac{\mathrm{cov}(\boldsymbol{s},\boldsymbol{t})}{\sigma_s\sigma_t} = \frac{E[(\boldsymbol{s}-\mu_s)(\boldsymbol{t}-\mu_t)]}{\sigma_s\sigma_t} \quad (2)$$

式中:$\boldsymbol{s}$ 为生成数据向量,$\boldsymbol{t}$ 为字符映射表中的向量。$\rho_{s,t}$是向量 $\boldsymbol{s}$、$\boldsymbol{t}$ 的协方差与标准差的商。值在[-1,1],绝对值越接近 1 越相关,越接近 0 越不相关。

将生成对抗网络输出的数据分别计算与各个字符的皮尔逊相关系数,取最相关的数据解码成字符,剔除其中不合理字符,然后组成域名。

2.2　域名生成模型

2.2.1　WGAN

在 GAN 模型中,生成器尽可能准确学习原始数据分布并生成类似数据,判别器作为一个二分类器尽可能区分真实数据和生成数据,二者在训练中不断提高自身的能力,最终达到纳什平衡,这正是生成对抗网络最大的特点。然而,在实际训练过程中,GAN 出现训练不稳定、梯度消失等问题。文献[22]对 GAN 出现的问题给出了理论的解释,即模型用于衡量数据相似性的损失函数 Jensen-Shannon 散度会出现常数,此时无法继续作出调整。WGAN 则优化了损失函数,用 Earth-Mover 距离替换 Jensen-Shannon 散度以衡量真假数据间的差距,同时将算法做了部分调整,优化了朴素 GAN 训练不稳定、梯度消失等问题[23]。

本文使用 WGAN 模型将目标函数描述为式(3)。

$$\min_G \max_D V(D,G) = E_{\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}[D(\boldsymbol{x})] + E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[1-D(G(\boldsymbol{z}))] \quad (3)$$

式(3)可以分成两个函数:最大化函数和最小化函数,分别如式(4)-式(5)。

$$\max_D V(D,G) = E_{\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}[D(\boldsymbol{x})] + E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[1-D(G(\boldsymbol{z}))] \quad (4)$$

式(4)是最大化函数。在训练过程中,对于判别器 D,输入真实的域名,$D(\boldsymbol{x})$的值越大越好。输入生成的域名,$D(\boldsymbol{x})$的值越小越好,前面加负号则变大,两个公式都变大,所以有 $\max_D V(D,G)$。

$$\min_G V(D,G) = E_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[1-D(G(\boldsymbol{z}))] \quad (5)$$

式(5)是最小化函数。对于生成器 G,生成的域名输入到判别器,判别器的输出越大越好,判别器尽可能认为数据是真实的。前面加个负号则越来越小,所以有 $\min_G V(D,G)$。

2.2.2　生成模型参数

经过反复训练调整,得到最终的 WGAN 网络模型:

生成网络:由输入层、2 层隐藏层和输出层组成。输入数据符合高斯分布的噪声,激活函数采用 ReLU 函数。隐藏层的节点分别是 200 和 150。输出层节点设置为 12 维,采用 tanh 激活函数。采用更能体现数据分布差异的 Earth-Mover 距离代替使用 GAN 模型的目标函数,推导之后体现在代码中是去掉 log,即式(5)。

判别网络:由输入层、2 层隐藏层和输出层组成。其输入数据一部分来自生成器生成的数据,一部分来自数据集中的真实数据,数据通过节点分别是 200 和 150 的隐藏层进行训练。输出层不采用 sigmoid 函数,真实数据和生成数据分别计算。目标函数为式(4)。

由于 Adam 算法会引起训练不稳定等问题,故本文采用 RMSProp 算法作为优化器。训练过程中,每更新 5 次判别网络,更新 1 次生成网络。学习率设为 0.000 2。此外,权值裁减至[-0.01,0.01]。

3 实验与分析

3.1 实验环境

本文使用的实验平台及配置信息如表 1 所示。

表 1 实验平台及配置

实验平台	配置
操作系统	Windows 10
内存	8 GB
CPU	Intel(R) Core(TM) i5-420M
编程语言	Python 3.7.2
深度学习框架	TensorFlow 1.13.1
机器学习平台	Weka 3.8

3.2 数据集

Conficker. C 恶意域名数据集和 Alexa 良性域名数据集是全球公认并使用较多的数据集。Alexa 会根据各个网站的链接数和用户访问量定期更新域名,越靠前的网站,相应的知名度也会越高,作为良性域名数据集更具有说服力。本文实验数据集选取近 50 万 Conficker. C 恶意域名,同时选取最新 Alexa 排名前 100 万的良性域名。前期 skip-gram 模型编码只使用 Conficker. C 恶意域名训练,后期分析生成数据有效性的过程中,使用生成恶意数据、Conficker. C 数据和 Alexa 数据。

DGA 算法主要针对的是二级域名,因此本文研究的主体是二级域名。例如 xinlang. com 中 xinlang 部分。实验前需要对数据集进行简单的处理,基于 python 语言,获取二级域名。

3.3 实验过程

本文目的为运用 skip-gram 与 WGAN 模型完成伪 DGA 域名的生成,达到对恶意域名级的扩充。因此实验的核心任务是域名的生成以及对生成域名的有效性、合理性验证。实验步骤如下:

1) 域名编码。skip-gram 训练 Conficker. C 数据集,将字符映射成对应的词向量,然后完成对域名数据集的编码工作。

2) 域名的生成。设计、训练 WGAN 读取步骤 1) 结果作为输入数据,每训练 5 次判别器,训练 1 次生成器。每迭代 500 次便输出一次数据进行解码,解码的域名存储在文本中。

3) 生成数据的有效性分析。针对生成恶意域名的有效性判定尤为重要,为验证实验结果的有效性,本文采用特征选择及机器学习相结合的方法进行验证:

(1) 特征选择:选用域名总长度、大小写字母及数字的数量、连词号的数量、n-gram 的频率($n=2,3,4$)、元音字母的频率、辅音字母的频率特征。

(2) 机器学习:选取随机树、随机森林和 J48 决策树三种分类算法。

4) 比对数据集划分。以下是三种数据集分类的描述。

(1) 首先用 Conficker. C 恶意域名正样本和 Alexa 的良性域名负样本通过上述步骤处理后用 Weka 平台进行分类,其结果作为后面分类的标准。

(2) 用 WGAN 模型生成的恶意域名正样本和 Alexa 的良性域名负样本,同样经过上述数据处理部分,选择相同的数据特征进行分类,其结果和(1)进行比较,从而判断生成样本的有效性。

(3) Conficker. C 恶意域名、生成恶意域名正样本和 Alexa 的良性域名负样本进行混淆分类,分类结果与(1)和(2)比较。

4 实验结果

4.1 编码结果分析

为了直观分析编码结果,本文运用 PCA 技术,将字符词向量维度降至 2 维,其空间距离代表字符间的相似度。如图 4 所示,字符按相似程度分别聚簇,其中字符'q'、'o'与右上角字符比左下角字符更聚集,相似度更高。相较于向量正交的 one-hot 编码(26 维),skip-gram 编码(4 维)的词向量维度低,降低计算复杂度,提升速度。

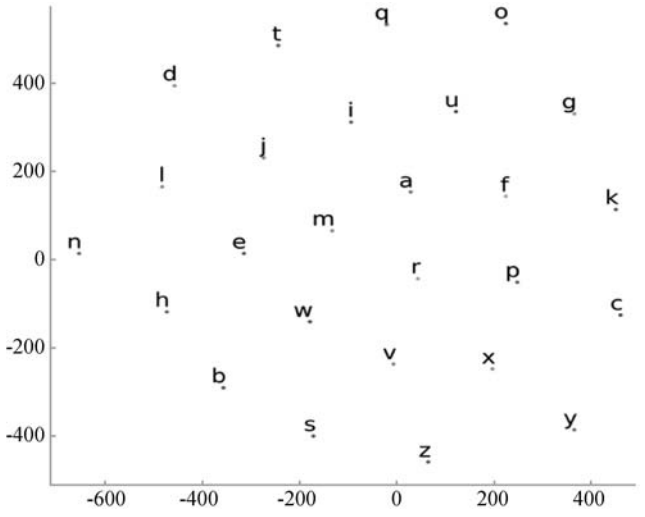

图 4 字符可视化映射

4.2　真实恶意域名与生成恶意域名表征对比分析

如图 5 所示,由于采用改进的 WGAN 模型,生成的恶意域名并没有出现字符大量重复的情况,从域名长度、字符间转换等表面特征看,生成的恶意域名与真实恶意域名相似度较高。

图 5　真实恶意域名与生成恶意域名示例

图 6 是从一元字符频率的角度来分析二者的区别,图中圆圈代表真实样本,星星代表生成样本。由图可知,真实样本的各个字符的频率在 0.04 范围内较小的波动。生成样本较真实样本存在波动,但其上下波动的幅度不大。故生成样本与真实样本具备一定的相似度。

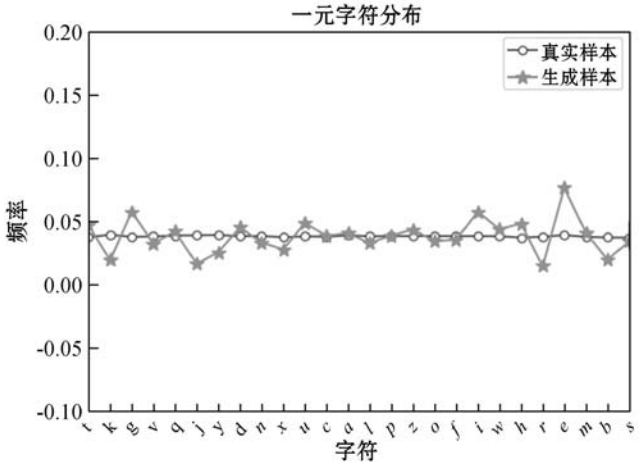

图 6　真实样本和生成样本的一元字符频率折线图

4.3　分类结果及分析

为验证生成域名的有效性,本文选取机器学习中的 J48 决策树、随机森林、随机树用于实验的对比分析算法。随机树通过 n 个特征的信息增益的最大节点迭代构造树和分类。随机森林是通过决策树的投票情况来分类,决策树是由随机数据集和特征集构成。J48 决策树根据分治策略,逐个加入特征进行分类。表 2、表 4、表 6、表 8、表 10 展示的是样本通过不同分类器的正确率、错误率、精确率、F-Measure、ROC 面积。表 3、表 5、表 7、表 9、表 11 展示的是分类正确错误数量以及模型的构建时间。

表 2　Conficker. C 正样本和 Alexa 负样本

分类器	正确率	错误率	精确率	F-M	ROC 面积
J48	0.994	0.006	0.994	0.994	0.996
随机树	0.997	0.003	0.997	0.997	0.997
随机森林	0.998	0.001	0.999	0.999	1

表 3　样本分类结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间/s
	正确样本	负样本	正样本	负样本	
J48	8451	9922	24	78	0.48
随机树	8445	9972	30	28	0.06
随机森林	8465	9990	10	10	3.17

表 2、表 3 是真实恶意域名与真实良性域名的分类结果。表中所有分类器的正确率都达到了 99% 以上。随机森林的正确率最高,它的构建模型时间也最长。表 2 的结果为下面各数据集的分类提供基准,同时可以表明特征选择的规则是有效的。

表 4　生成的正样本和 Alexa 样本(1∶1)

分类器	正确率	错误率	精确率	F-M	ROC 面积
J48	0.951	0.048	0.953	0.951	0.972
随机树	0.983	0.017	0.983	0.983	0.983
随机森林	0.981	0.019	0.981	0.981	0.981

表 5　样本分类的结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间/s
	正确样本	负样本	正样本	负样本	
J48	9720	9276	245	724	0.44
随机树	9812	9815	153	185	0.08
随机森林	9903	9879	208	167	4.85

表 6　生成的正样本和 Alexa 样本(1∶2)

分类器	正确率	错误率	精确率	F-M	ROC 面积
J48	0.945	0.055	0.946	0.946	0.967
随机树	0.980	0.020	0.980	0.980	0.978
随机森林	0.989	0.011	0.989	0.989	0.983

表 7　样本分类的结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间/s
	正确样本	负样本	正样本	负样本	
J48	4737	9430	261	569	1.85
随机树	4849	9852	149	147	0.13
随机森林	4941	9894	57	105	9.6

表4、表5是生成的恶意域名和良性域名的分类结果,分类器和提取的特征同表2、表3。从表中可以看出,三个分类器的正确率都比较高,尤其是随机树和随机森林达到了98%以上。而表6、表7在上述基础上,样本总数减少且比例由原来1∶1变成1∶2,仍呈现出较好的分类结果。说明在选取相同特征的前提下,生成的恶意域名可以充当真实恶意域名。

表8 混淆样本(1∶1)和Alexa分类结果

分类器	正确率	错误率	精确率	F-M	ROC面积
J48	0.963	0.037	0.966	0.963	0.979
随机树	0.987	0.013	0.987	0.987	0.997
随机森林	0.992	0.008	0.992	0.992	0.998

表9 样本分类的结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间/s
	正确样本	负样本	正样本	负样本	
J48	8830	7548	170	452	0.34
随机树	8915	7867	85	133	0.17
随机森林	8967	7903	33	97	3.13

表10 混淆样本(2∶1)和Alexa分类结果

分类器	正确率	错误率	精确率	F-M	ROC面积
J48	0.963	0.036	0.964	0.963	0.973
随机树	0.977	0.023	0.977	0.977	0.974
随机森林	0.988	0.012	0.988	0.988	0.992

表11 样本分类的结果及模型构建时间

分类器	分类正确数		分类错误数		构建时间/s
	正确样本	负样本	正样本	负样本	
J48	19612	9297	860	234	1.87
随机树	19515	9805	352	331	0.19
随机森林	19780	9856	301	66	9.06

表8、表9是真实恶意域名、生成恶意域名作为正样本和Alexa域名作为负样本进行分类,结果显示随机树、随机森林都达到98%以上,J48表现的稍微逊色一些,但是也达到96%。表10、表11将混淆样本中生成恶意域名和真实域名的比例由1∶1变成2∶1,同时样本总数增加且比例由1∶1变成2∶1,分类正确率依旧较高。说明生成的恶意域名能够较好的隐藏在真实恶意域名中不被区分出来。生成的恶意域名可以作为真实恶意域名进行机器学习的检测和训练。

综上三组对比实验,以第一组为基准,后两组将生成的恶意域名分别与良性域名、混淆域名作分类进行横向对比,同时每组实验中分别添加样本数及比例不同的分类实验进行纵向对比,结果均表现出较高的分类正确率,说明了生成恶意域名的有效性。

5 结 语

恶意域名识别问题一直是网络安全领域的重要研究点。本文提出skip-gram和WGAN的结合模型,能够生成和预测DGA域名,并通过实验验证此方法的合理性。扩充后的数据集不仅可以丰富黑名单系统内的域名,也可以作为域名检测算法的训练样本。下一步研究工作将继续放在合适的编码和解码上,同时研究如何优化生成对抗网络模型以达到更好的效果。

参 考 文 献

[1] Mockapetris P. Domain Names—Concepts and Facilities. [EB/OL]. [2019-11-06]. http://ftp.sjtu.edu.cn/pub/rfc/pdfrfc/rfc1034.txt.pdf.

[2] Stone-Gross B, Cova M, Gilbert B, et al. Analysis of a botnet takeover[J]. IEEE Security & Privacy, 2011, 9(1):64-72.

[3] 江健,诸葛建伟,段海新,等. 僵尸网络机理与防御技术[J]. 软件学报,2012,23(1):82-96.

[4] 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(9):310-316.

[5] Schiavoni S, Maggi F, Cavallaro L, et al. Phoenix: DGA-based botnet tracking and intelligence[C]//International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer,2014.

[6] Mowbray M, Hagen J. Finding Domain-generation algorithms by looking at length distribution[C]//IEEE International Symposium on Software Reliability Engineering Workshops. IEEE, 2014.

[7] Yadav S, Reddy A K K, Reddy A L N, et al. Detecting algorithmically generated malicious domain names[C]//Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. ACM, 2010:48-61.

[8] Wang W, Shirley K. Breaking bad: Detecting malicious domains using word segmentation[EB]. arXiv: 1506.04111, 2015.

[9] Truong D T, Cheng G. Detecting domain-flux botnet based on DNS traffic features in managed network[J]. Security and Communication Networks, 2016,9(14):2338-2347.

[10] 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.

[11] Zhou Y L, Li Q S, Miao Q D, et al. DGA-based botnet detection using DNS traffic[J]. Journal of Internet Services and Information Security, 2013, 3(3/4):116 – 123.

[12] Yu B, Gray D L, Pan J, et al. Inline DGA detection with deep networks[C]//2017 IEEE International Conference on Data Mining Workshops(ICDMW). IEEE Computer Society, 2017.

[13] Anderson H S, Woodbridge J, Filar B. DeepDGA: Adversarially-tuned domain generation and detection[EB]. arXiv: 1610.01969,2016.

[14] Zheng S, Bao H, Xu J, et al. A bidirectional hierarchical skip-gram model for text topic embedding[C]//2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016.

[15] 覃玉冰,邓春林,杨柳. 基于皮尔逊相关系数的网络舆情评估指标体系构建研究[J]. 情报探索,2018,252(10):19 – 23.

[16] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//International Conference on Neural Information Processing Systems,2014.

[17] 许春冬,许瑞龙,周静. 基于自动编码生成对抗网络的语音增强算法[J]. 计算机工程与设计,2019,40(9):2578 – 2583.

[18] 蒋鹏飞,魏松杰. 基于深度森林与 CWGAN-GP 的移动应用网络行为分类与评估[J]. 计算机科学,2020,47(1):287 – 292.

[19] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality [J]. Advances in Neural Information Processing Systems, 2013, 26:3111 – 3119.

[20] Lai S, Liu K, He S, et al. How to generate a good word embedding? [J]. IEEE Intelligent Systems, 2016, 31(6): 5 – 14.

[21] 张玉英. 一种基于加权欧氏距离聚类方法的研究[J]. 计算机应用,2006,26(S2):152 – 153.

[22] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks [EB]. arXiv: 1701.04862,2017.

[23] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN[EB]. arXiv:1701.07875,2017.

(上接第 222 页)

参 考 文 献

[1] Dumitrescu A, Tóth C D. The traveling salesman problem for lines, balls, and planes [J]. ACM Transactions on Algorithms,2013,12(3):1 – 29.

[2] Zhang W, Zhang E, Zheng F. Online two stage k-Search problem and its competitive analysis[J]. International Journal of Foundations of Computer Science,2016,27(6):653 – 663.

[3] Tan X, Jiang B. Optimum sweeps of simple polygons with two guards[J]. Information Processing Letters, 2014, 114(3): 130 – 136.

[4] Dumitrescu A, Mitchell J, Żyliński P. Watchman routes for lines and line segments[J]. Computational Geometry Volume,2014,47(4):527 – 538.

[5] CzyzowiczJ, Labourel A, Pelc A. Optimality and competitiveness of exploring polygons by mobile robots[J]. Information and Computation,2011,209(1): 74 – 88.

[6] Icking C, Kamphans T, Klein R, et al. Exploring simple grid polygons[C]//International Computing and Combinatorics Conference. Springer,2005:524 – 533.

[7] Zhang G, Cheng Y, Qin L, et al. An improved online evacuation strategy from a convex region on grid networks[J]. Journal of Combinatorial Optimization, 2018,36(2):44 – 54.

[8] Kolenderska A, Kosowski A, Matafiejski M, et al. An improved strategy for exploring a grid polygon[C]//International Conference on Structural Information&Communication Complexity. Springer,2009.

[9] Keshavarz-Kohjerdi F, Bagheri A. A linear-time algorithm for finding Hamiltonian (s, t)-paths in odd-sized rectangular grid graphs with a rectangular hole[J]. Journal of Supercomputing,2017,73(9):3821 – 3860.

[10] Hoffmann F, Icking C, Klein R, et al. The polygon exploration problem[J]. SIAM Journal on Computing, 2001,31(2):577 – 600.

[11] Icking C, Kamphans T, Klein R, et al. Exploring grid polygons online[EB]. arXiv:1012.5240,2010.

[12] Ghosh S K, Burdick J W, Bhattacharya A, et al. Online algorithms with discrete visibility-exploring unknown polygonal environments[J]. Robotics& Automation Magazine IEEE, 2008,15(2):67 – 76.

[13] Megow N, Mehlhorn K, Schweitzer P. Online graph exploration: New results on old and new algorithms[C]//Proceedings of the 38th international conference on Automata, languages and programming. ACM,2012:478 – 489.

[14] Ghosh S K, Klein R. Online algorithms for searching and exploration in the plane[J]. Computer Science Review,2010,4(4):189 – 201.

[15] Herrmann D, Kamphans T, Langetepe E. Exploring simple triangular and hexagonal grid polygons online[EB]. arXiv: 1012.5253,2010.