DNTrans: 基于 Transformer 的黑产域名变换生成方法

王 博, 施 凡

国防科技大学电子对抗学院 合肥 中国 230031

摘要 目前很多黑产团伙为了对抗监管部门对其网站域名的封锁, 会批量搭建黑产网站和注册大量的黑产域名。这些批量注册的域名之间存在着一定的相似性, 这种相似性使得研究者可以对已知域名进行分析, 进而研究未知域名的生成。本文研究区别于以往研究对黑产域名生成的处理方法, 我们将域名生成任务转变为翻译任务。首先, 我们采用 Bi-LSTM 将训练数据中的域名转为表征向量后进行层次聚类, 将相似的域名聚为一类, 并且通过参数设置, 使得聚类簇中的域名个数尽量分布均匀。然后, 根据聚类结果生成翻译模型所需的域名对数据。最后, 使用 Transformer 模型自动学习相似域名之间潜在的变化规则进行黑产域名的变换生成。其中域名生成结果检验采用的是我们自己提出的两阶段黑产网站检测模型, 模型通过设置置信度阈值的方式控制检测模型大小以及所需数据来平衡识别准确率和效率。实验表明, 生成算法生成的域名中, 可访问域名中黑产域名比率为 19.1%, 黑产域名的扩展倍数达到了 359.98, 即通过一个黑产域名可以平均扩展出近 360 个新的黑产域名。实验结果证明了该方法在黑产域名变换生成上的有效性, 并解决了现有公害域名生成方法难以控制域名生成的范围, 存在大量的无效域名的问题。

关键词 域名生成算法; 翻译模型; 聚类; 多模态

中图法分类号 TP391 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2025.03.11

DNTrans: Illicit Domain Name Transformation Generation Method Based on Transformer

WANG Bo, SHI Fan

College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China

Abstract Currently, many illicit groups, in order to evade regulatory authorities' domain name blocking efforts, engage in the mass creation of illegal websites and register a substantial number of illicit domain names. These bulk-registered domain names exhibit a certain level of similarity, a resemblance that allows researchers to analyze known domain names and subsequently explore the generation of unfamiliar domain names. This paper's research approach sets it apart from previous studies on the generation of illicit domain names. We convert the domain name generation task into a translation task. Firstly, we employ a Bi-LSTM to convert domain names in the training data into representation vectors, and then perform hierarchical clustering to group similar domain names together. Additionally, through parameter settings, we aim to evenly distribute the number of domain names within each cluster. Subsequently, based on the clustering outcomes, we generate the domain name pairs essential for training the translation model. Finally, we utilize a Transformer model to automatically grasp the latent alteration patterns between similar domain names, thus generating transformed versions of illicit domain names. The assessment of domain name generation results incorporates our self-devised two-stage detection model for illicit websites. The illegal website detection model controls the size of the detection model and the required data by setting confidence thresholds to balance recognition accuracy and efficiency. Experimental results demonstrate that among the domain names generated by the algorithm, the proportion of accessible illicit domain names is 19.1%, and the expansion factor of illicit domain names reaches 359.98. This implies that, on average, nearly 360 new illicit domain names can be spawned by altering a single illicit domain name. The experimental results demonstrate the effectiveness of this method in generating transformations for illicit domain names, addressing the challenges associated with controlling the scope of domain name generation and the presence of numerous invalid domain names in existing illicit domain name generation methods.

Key words domain name generation algorithm; translation model; clustering; multimodal

通讯作者: 施凡, 副教授, Email: shifan17@nudt.edu.cn。

本课题得到国家重点研发计划项目(No. 2021YFB3100500)资助。

收稿日期: 2023-09-01; 修改日期: 2023-10-15; 定稿日期: 2025-01-22

1 引言

随着信息技术的发展, 传统的赌博、色情等黑色产业也开始转到线上。线上的这些黑产网站由于易被更广泛的人群接触到, 从而对整个社会造成了更大的危害。赌博和色情等黑产网站不仅对经济、社会和个人层面带来了巨大的危害, 还对整体的网络生态和用户安全构成了严重威胁。

现有的黑产网站检测主要集中于对网站的部分属性进行检测, 以确认其是否为黑产网站。这种审核性质的方法对于黑产网站的治理很重要, 但是这种方法对黑产网站的治理具有一定的滞后性, 不能主动的去探查一些未知的黑产网站。

域名的作用类似于互联网上的门牌号码, 它们引导着用户进入特定的网站。因此有效的找到潜在非法网站的域名是进行黑产网站治理的第一步。结合目前黑产网站经常批量注册大量相似域名的特点, 我们提出了一种黑产域名变换方法 DNTrans 以发现更多的黑产网站。

本方法旨在通过对现有黑产网站域名进行一定的变化, 以生成更多的黑产网站域名。首先, 我们使用双向循环神经网络对域名数据集进行特征学习, 从而获得它们的表征(第 3.1.1 节)。接下来, 采用层次聚类方法对相似的域名进行聚类(第 3.1.2 节), 以作为后续生成训练变换模型的数据来源。然后, 我们使用 Transformer 模型来学习相似域名对之间的转换规则, 从而实现输入已知黑产网站域名后, 输出大量潜在黑产网站域名的效果(第 3.4 节)。并且为了高效准确的确认生成的域名是否为黑产网站, 我们提出了一种两阶段黑产网站检测方法, 具体细节在第 3.2 节中介绍。

简而言之, 本研究在探索黑产网站发现方面做出了以下贡献:

(1) 首次将翻译的思想应用到域名生成领域, 将相似的域名对认为是不同的语言, 借助 Transformer 的模型实现未知黑产网站域名的发现。基于一个黑产域名, 模型可以平均扩展出 360 个新的黑产域名。

(2) 提出了一个两阶段的黑产网站检测模型, 可以在高准确率的情况下, 快速检测一个网站是否为黑产网站, 用于检测本文通过域名变换发现的存活网站是否为黑产网站。

(3) 我们将生成的有效黑产域名做成了一个黑产网站域名数据集, 数据集开源在 https://www.kaggle.com/datasets/listone/illegal-domain, 一共包含 35998 条域名数据。我们希望更多的研究人员可以在这个数据集的基础上发现更有效的黑产网站域名生成方案。

综上所述, 本文的目的是通过已发现的黑产网站去尽可能发现更多未知的黑产网站, 由点及面, 通过一个黑产网站能够拓展出多个与之相关的黑产网站, 统一进行治理。

2 研究背景及相关工作

2.1 研究背景

2.1.1 黑产域名分布特征

很多黑产网站为了防止被监管封锁, 会选择批量注册域名。如图 1 所示, 有 5 个不同的二级域名共用同一个 SSL 证书, 这 5 个域名有很明显的共同特征, 即全部都由数字构成。也有很多没有共用同一个 SSL 证书, 但是其组成结构非常相似, 如图 2 所示, 这四个域名都为“cjnh”和数字构成, 这四个域名之间可以通过一定的规则进行转换。我们所提的基于 Transformer 模型的方法就是为了能自动化挖掘出类似这种的转换规则。

除了这种黑产域名的相似性之外, 黑产域名与正常域名的字符分布与长度分布也有着较大差异。我们将域名中的二级域名 SLD 提取出来做了统计分析对比, 如图 3a 所示, 这个图分别统计了 27 万黑产域名和 Alexa top 1m 中前 27 万条正常域名的字符分布。纵轴是每个字符, 横轴是所有二级域名中每个字符的个数和, 为了方便展示, 每个坐标值除以 10000 作为横坐标。通过图 3a 可以明显看出黑产域名和正常域名的差别, 黑产域名字符分布较均匀, 数字占比很大, 而正常域名数字占比很少, 几乎都是字母, 并且也存在高频字母和低频字母之分, 这也符合正常的字符分布。我们也对正常域名和黑产域名的二级域名长度做了统计, 如图 3b 所示, 纵轴表示二级域名的个数, 横轴表示二级域名的长度。可以看到正常域名的曲线比较平缓, 而黑产域名的曲线很尖锐, 域名长度分布较为集中。

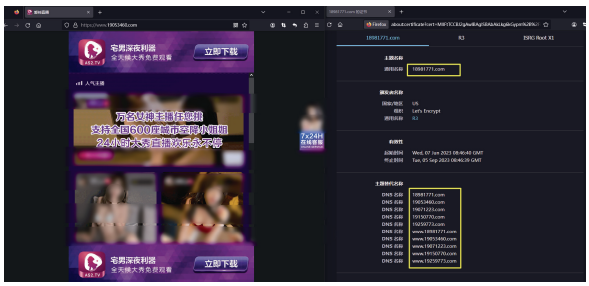

图 1 共用 SSL 证书示例

Figure 1 Shared SSL certificate example

图 2　黑产网站域名相似性(a) cjnh999.com;(b) www.cjnh8899.com;(c) www.cjnh1122.com;(d) www.cjnh8588.com

Figure 2　Illicit website domain name similarity. (a) cjnh999.com;(b) www.cjnh8899.com;(c) www.cjnh1122.com;(d) www.cjnh8588.com

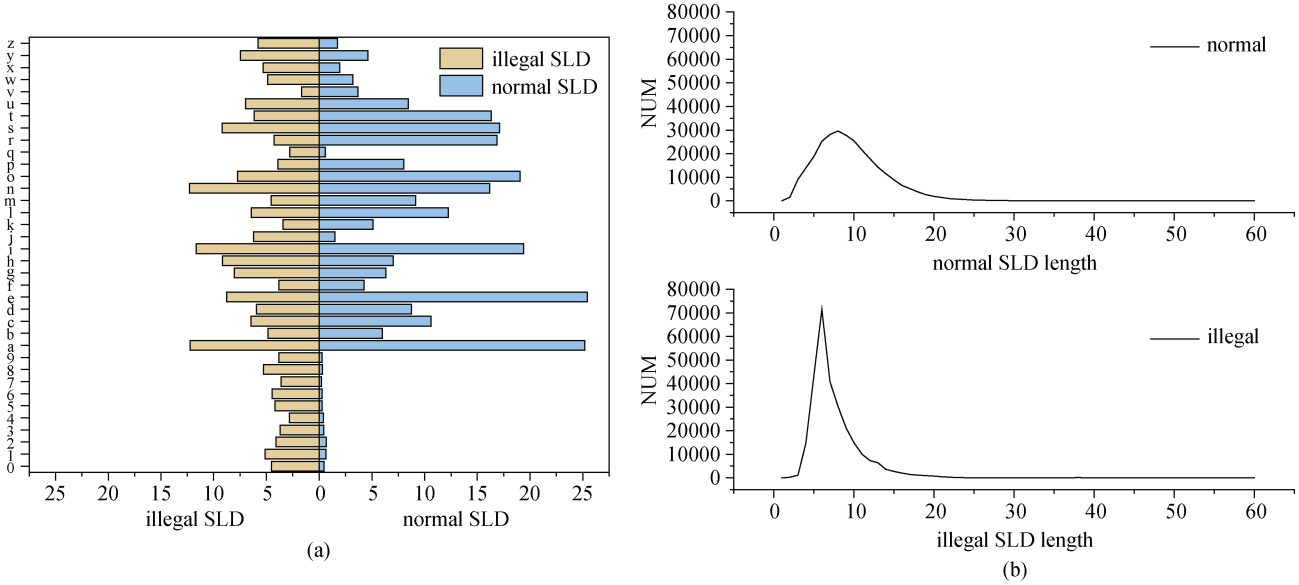

图 3　黑产域名与正常域名之间的差异性

Figure 3　Differences between illicit domain names and normal domain names

总的来说, 黑产域名的结构相似性以及这种与正常域名之间的分布差异为通过变换黑产域名来生成新的黑产域名提供了实践依据。

2.1.2　黑产网站的伪装

随着监管与黑产网站的对抗, 目前很多黑产网站会采用一些伪装手段来躲避一些自动化程序的监管, 包括:

(1) 变体文字: 存在用中英文替换、数字替换文字、特殊符号替换文字以及艺术字体等, 这些手段会导致在正常文字上训练得到的文本分类模型失效;

(2) 对抗图像: 这种方式目前遇到较少, 会在网页图像上加入不同颜色的杂乱符号、模糊等, 这些手段会导致一些图像分类模型失效;

(3) 验证码: 有些网页会设置验证码, 只有输入正确验证码才能访问到真实的网页, 这种方法会导致没有配备验证码识别的网络爬虫难以获取真实网页内容;

(4) 网页 title 伪装: 这种伪装方式比较简单, 网页内容是黑产网站的内容, 只将网页的 title 部分修改为正常内容, 这种手段对于仅对网页 title 进行检测的模型有效;

(5) 图片构建网页: 这种方式仅采用图片构建网页内容, 网页 HTML 中仅有<img>标签对图片的引用, 这种方式会使仅对 HTML 进行检测的模型失效。

通过对很多黑产网站的观察, 发现网页 title 伪装和图片构建网页伪装比较常见。如图 4 所示两个网页截屏都为黑产网站, 其中图 4a 将 title 进行了伪装, 网页的 title 被替换为了一个公司的名称“朝阳眉雀电子有限公司”。图 4b 显示的网页全部由图片构成, 在网页 HTML 源码中并不存在任何有区分性的文本, 网页图片上的文本“真人真钱真娱乐, 棋牌体育”, 这个文本对于区分赌博网站和正常网站是很有效的信息。

在这些手段的伪装之下, 我们很难只通过网页的一种属性来实现高精度的识别。而从网页的多维度特征去综合判断网页是否为黑产网站会导致识别

(a) (b)

图 4 网站网页伪装

Figure 4 Website page camouflage

效率下降。因此, 在本篇文章中我们提出了一种两阶段黑产网站检测方法, 通过两个阶段的组合来平衡识别效率和识别准确率。第一阶段使用获取快的网页 HTML 进行识别, 如果识别置信度达到预设阈值就直接输出判别结果, 结束检测。否则, 就将文本表征传入第二阶段和网页截屏的表征融合后进行判别, 最终得到结果。

2.2 相关工作

2.2.1 黑产网站检测

目前, 有许多工作从不同视角出发进行黑产网站的识别, 主要可以分为五类: 基于黑名单的、基于图(graph)的、基于 URL 的、基于单一网页内容的、基于混合网页内容的。

基于黑名单的方法。基于黑名单的方法由于黑产网站不断变化, 也提出了一些变种。Prakash 等[1]考虑到对黑名单条目的精确匹配使得黑产网站很容易规避, 因此提出了一种启发式组合方法, 将已知的黑产网站域名进行简单组合以发现新的黑产网站域名。Akiyama 等人[2]也采用了类似的方法。他们假设未知恶意 URL 存在于由同一黑产网站团伙创建的已知恶意 URL 附近, 因此提出了一种黑名单 URL 生成方法来扩充已有黑名单。Rao 等人[3]将原始基于域名的黑名单修改为基于 Simhash 的黑名单。他们为每个网站生成一个 Simhash, 通过 Simhash 的匹配实现对黑产网站的发现。

基于图的方法。基于图的方法主要将网站与网站之间的联系、网站内部节点之间的联系等用图来描述, 通过节点之间的关系来实现对黑产网站的识别。Wenyin 等人[4]将与可疑网页有直接或间接关联的网页提供给用户, 由用户来确定可疑网页是否是钓鱼网页。这种方法不能主动识别网站, 只是为用户提供了参考, 最终由用户来进行判断。Futai 等人[5]分析了来自 ISP 的真实 IP 流, 提出了一种基于图挖掘和置信传播的方法, 并且实现了分布式功能, 提高了分析识别黑产网站的效率。Zhao 等人[6]提出了一种使用对比学习检测色情网站的健壮的端到端框架 Porn2Vec, 他们通过网站、网页、图像、文本及其交互关系构成异构图对色情网站进行建模, 将色情网站检测任务转变为异构图节点分类任务, 通过多种数据的表示提高了模型的鲁棒性。Li 等人[7]使用异构信息网络 (HIN)对候选赌博和色情页面和页面上的资源进行建模。然后将分类算法应用于 HIN 以识别赌博和色情页面。

基于 URL 的方法。基于 URL 的方法主要是从 URL 中提取特征来实现分类。Ma 等人[8]使用统计的方法来发现恶意 URL 的明显词汇特征和基于主机的属性特征, 通过这些特征来学习预测模型。Yuan 等人[9]将字符嵌入与 URL 的结构相结合以获得 URL 的向量表示。在 URL 的向量表示上使用现有的分类算法实现对黑产网站的识别。Zhu 等人[10]通过决策树和局部搜索方法进行最优特征选择, 防止神经网络分类器过拟合, 提高黑产网站的识别性能。Yan 等人[11]提出了一种学习 URL 嵌入的无监督学习算法, 通过学习一个良好的 URL 表征来实现对黑产网站的检测。Yang 等人[12]首先使用字符嵌入技术将 URL 转为固定大小的矩阵, 然后使用卷积神经网络和随机森林实现对黑产网站的识别。Sun 等人[13]提出了基于证明和文本分析的分类方法, 通过 Bert 模型对赌博域名和正常域名进行分类。Su 等人[14]通过对域名解析记录进行分析, 建立了混合临时随机域名过滤算法, 使得域名服务器可以及时阻止对黑产网站的访问。Min 等人[15]通过分析垃圾邮件 URL 的特征来识别移动端在线赌博网站。

基于单一网页内容的方法。基于单一网页内容的方法主要是通过网页中的一种属性对黑产网站进行识别。Li 等人[16]引入视觉特征来识别赌博网站和

色情网站, 通过 BoW 模型选择有效特征来识别赌博网站和色情网站的截图。Liu 等人[17]从用户的角度出发, 使用卷积神经网络进行恶意网站的识别。Jain 等人[18]通过分析网站 HTML 源代码中的超链接来检测网络钓鱼网站, 将超链接的特定特征分为 12 个不同的类别, 并使用这些特征来训练机器学习算法。Cernica 等人[19]基于计算机视觉的框架来查找钓鱼网页, 通过计算网站截屏之间的相似度进行钓鱼网站的识别。

基于混合网页内容的方法。基于混合网页内容的方法结合了网页的网页截屏、HTML 以及 JavaScript 代码等多种属性来进行黑产网站的识别。Zhang 等人[20]使用由基于 URL、基于 Web、基于规则和基于文本内容的特征组成的混合特征, 通过 ELM 构建网页分类模型。Zhu 等人[21]从域名、HTML、JavaScript 中提取特征, 为了减少冗余特征, 引入了一个新的指标, 即特征有效性值, 以评估敏感特征对网络钓鱼网站检测的影响。Yang 等人[22]将 URL 统计特征、网页代码特征、网页文本特征组合成多维特征后传入神经网络进行黑产网站的识别。Chen 等人[23]通过逻辑回归融合基于视觉的分类器和基于文本内容的分类器的分类结果来得到最终的预测结果。Gandotra 等[24]基于网页、URL 和 HTML 进行单独分类。然后将所有特征进行集成用于分类。Xiong 等人[25]根据网页文本内容、访问路径、网站关联平台属性构建网站特征向量, 然后根据特征向量与已发现黑产网站的相似度来进行黑产网站的识别。Liu 等人[26]在不同尺度上进行语义信息的融合, 提出了三种不同深度的融合模型, 三种模型与视觉和文本方法相比性能有所提升。

总的来说, 基于黑名单的方法由于黑产网站的不断变化, 导致效果不佳。基于图的方法考虑到了不同网站间以及站点内部的关系, 提高了黑产网站识别的鲁棒性, 但是基于图的方法比较复杂, 效率较低。基于 URL 的方法处理简单, 速度也较快, 但是 URL 中有效的信息较少, 导致整体识别准确率不高。基于单一网页内容的方法在与黑产网站对抗的初期比较有效, 但现在随着黑产网站的伪装手段不断变多, 单一特征很容易被绕过检测。基于混合网页内容的方法被越来越多的研究人员采用。通过对不同属性的特征进行组合可以提高模型检测的鲁棒性, 但是也会存在多个模型组合导致执行效率低下的问题。因此有必要研究如何优化检测流程, 既能解决单一网页内容容易被绕过的问题, 又可以解决多个属性融合导致的检测效率低下的问题。

2.2.2 黑产域名的生成

目前关于黑产域名的生成研究主要应用在僵尸网络中的 DGA 算法, 这些传统的域名生成算法在域名生成的过程中会遵循一定的算法机制。例如 Bamital[27]和 Dyre[28]使用哈希的十六进制表示来生成 AGD(Algorithmically Generated Domain, 算法生成域名), 并将时间源合并到 AGD 计算中。Conficker[29]将系统时间、线程 id 和进程 id 合并到伪随机生成器中。Pushdo[30]计算种子的 MD5 散列的十六进制表示来生成域名。也有基于字典的 DGA, 例如 Matsnu[31]和 Suppobox[32]考虑到了许多合法域名包含至少一个自然语言单词的事实。因此, 通过连接从特定字典中随机选择的单词来生成域名, 让生成的恶意域名有合法域名的一些字符级特征。

上述的 DGA 生成的域名是根据预先约定的规则进行生成, 与为主动发现恶意域名的目的有所出入。现在针对恶意域名的主动生成发现的方法主要是采用 GAN 的方法进行生成。例如 Gong 等人[33]通过 GAN 学习假冒域名的特征, 然后生成潜在的假冒域名。Liang 等人[34]通过 GAN 生成赌博、色情域名。Pham 等人[35]通过 GAN 生成钓鱼网站域名。Valentim 等人[36]采用 GAN 生成通用的 URL, 可以用到域名抢注预测。Zhai 等人[37]使用 GAN 来模拟传统的 DGA 算法, 并且加强了抗检测性来抵抗 DGA 检测算法的检测。这些基于 GAN 的方法在生成阶段时是从训练阶段训练好的分布中采样随机变量后, 从随机变量生成的域名。这些生成方法难以控制生成的范围, 存在大量的无效的域名。

还有采用循环神经网络进行域名生成[38-39], 循环神经网络在训练阶段会学习到训练集的特征, 在生成阶段时由于参数固定, 输入开头字符后, 其输出相对固定。生成域名时需要遍历整个域名空间, 计算每个域名的概率, 舍弃较小概率的域名。这种方式也存在和 GAN 相似的问题, 输出的域名难以控制范围, 存在大量无效的域名。

因此, 我们首次将翻译的思想应用到域名生成领域, 通过输入已确认为黑产网站的域名后, 模型将输入的域名经过一定的变换后生成潜在的黑产网站域名, 这种方式不仅在训练阶段学习到了黑产网站域名的特征, 而且在生成阶段也可以通过有效的黑产域名输入来控制模型的输出域名范围。

3 方法设计

在本节中, 我们设计了基于 Transformer[40]的黑产域名变换生成方法, 如图 5 所示。我们的方法主要包括三个阶段: 构建数据集; 域名变换生成; 检验黑产域名。

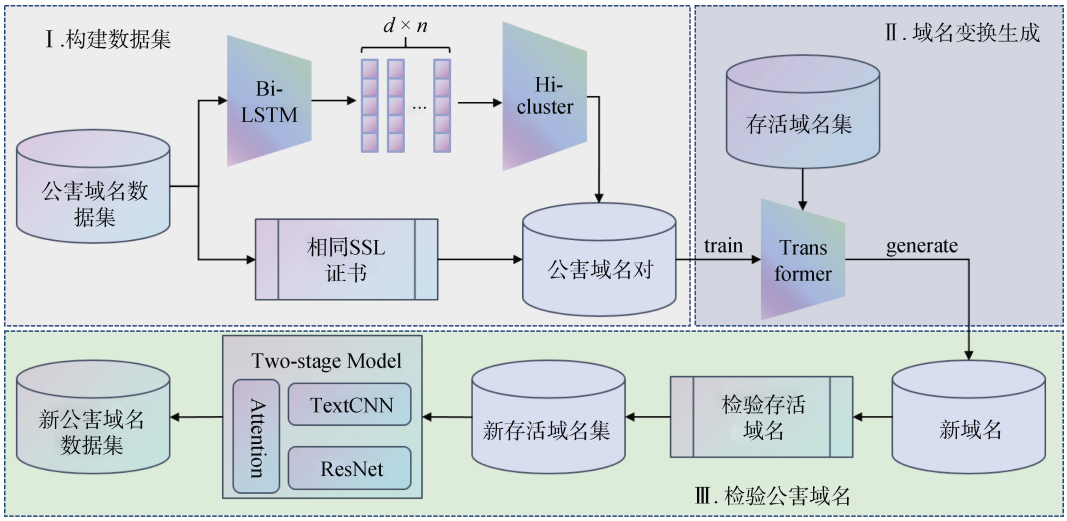

图 5　基于 Transformer 的黑产域名变换生成方法

Figure 5　Transformer-Based Illicit domain name transformation generation method

3.1　域名表征聚类

为了构造适合 Transformer 模型的训练数据, 我们采用了基于 Bi-LSTM 的域名表征模型和层次聚类模型。通过表征模型获取每个域名的表征向量, 之后将表征向量传入层次聚类模型获取 N 个聚类簇, 如图 6 所示。

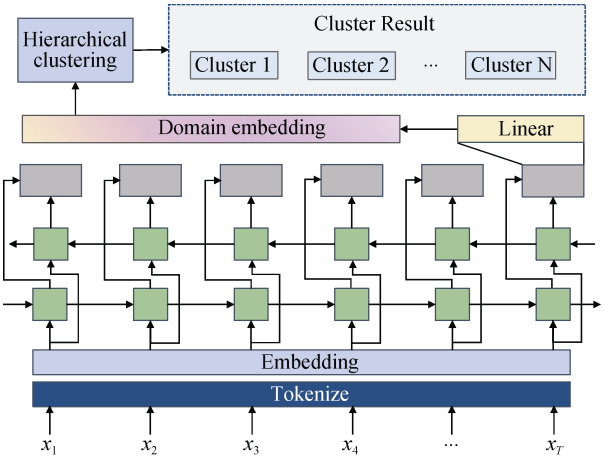

图 6 域名表征聚类模型

Figure 6　Domain representation clustering model

3.1.1　域名表征模型

循环神经网络在提出之初就是为了处理序列数据, 域名也可以看作是一种序列数据。我们构造了一个基于 Bi-LSTM 的表征模型, 表征模型采用无监督的方式进行训练。我们采用预测下一个字符的方式进行训练。图 6 中显示了基于 Bi-LSTM 的域名表征模型结构。首先我们将域名进行 tokenize 操作, 将域名中每个字符分开, 分为 $[x_1, x_2, \cdots, x_T]$, 并转为每个字符的索引 $[id_1, id_2, \cdots, id_T]$, 经过 Embedding 层后将字符 id 嵌入为向量。使用 Bi-LSTM 学习到域名中上下文特征, 并将前向 LSTM 和后向 LSTM 的输出在最后时刻进行拼接后输入到全连接层后将向量维度变为我们需要的域名表征向量维度大小。

由于组成域名的字符有限, 因此在模型中的 Embedding 层中将每个字符变为 one-hot 向量。在模型中, 我们采用了两层 Bi-LSTM 模型, 每层有 128 个神经元。在最后将前向 LSTM 和后向 LSTM 的输出拼接后得到 256 维向量, 之后通过全连接层将 256 维向量降为 64 维向量。这个 64 维向量就为最终的域名表征向量。

3.1.2　基于域名表征的层次聚类

在本文中, 我们使用层次聚类算法对域名进行聚类, 并将第 3.1.1 节中获得的表征向量相似的域名聚为一类。层次聚类的基本思想如下。

(i) 初始化: 将每个域名视为一个独立的类别。

(ii) 计算相似度矩阵: 计算每对域名之间的相似度, 可以使用欧几里德距离、余弦相似度等度量方式。

(iii) 合并最相似的类别: 从相似度矩阵中选择最相似的两个类别, 并将它们合并为一个新的类别。

(iv) 更新相似度矩阵: 更新相似度矩阵, 以反映新的类别之间的相似度。可以使用不同的聚类准则, 如单链接、完全链接、平均链接等。

(v) 重复步骤 (iii) 和 (iv) 直到满足某个停止准则, 例如达到指定的类别数量或相似度低于某个阈值。

整个层次聚类过程可以表示为一棵树状图, 被称为聚类树或谱系图。每个叶子节点代表一个单独的域名, 而内部节点表示合并的类别。最终, 我们可以根据聚类树的结构和相关阈值来确定最终的域名

聚类结果。可以根据需要选择树中的某个切割点，将树划分为不同的聚类簇。

我们对比了多种聚类算法，例如 K-means 算法、DBSCAN 算法、SCAN[41]算法等。我们最终选择层次聚类算法的原因主要有两点。第一，层次聚类可以不指定类别数，我们只需要指定 distance 参数即可，这对于无法预知类别数目的域名数据来说具有非常大的吸引力；第二，层次聚类获得的结果较为稳定，每个类别中的样本数目分布较均匀，这对于后面构造训练数据集非常有帮助。具体的对比结果见第 4.1 节。

3.2 基于图文融合的两阶段黑产网站检测

在本节中，我们设计了基于多模态数据融合的两阶段黑产网站高效识别方法。图 7 展示了用于黑产网站识别的模型结构。模型使用了两种模态数据，其中文本数据为网页 HTML。文本占用空间小，具有加载速度快的优势。因此在我们的方法中模型首先会使用文本进行判断，如果文本判断的结果置信度超过我们预设的阈值，此时会直接输出结果，不再进行图像模型的判断。否则，会将文本表征与图像表征进行融合后一起判断，最终得到结果。

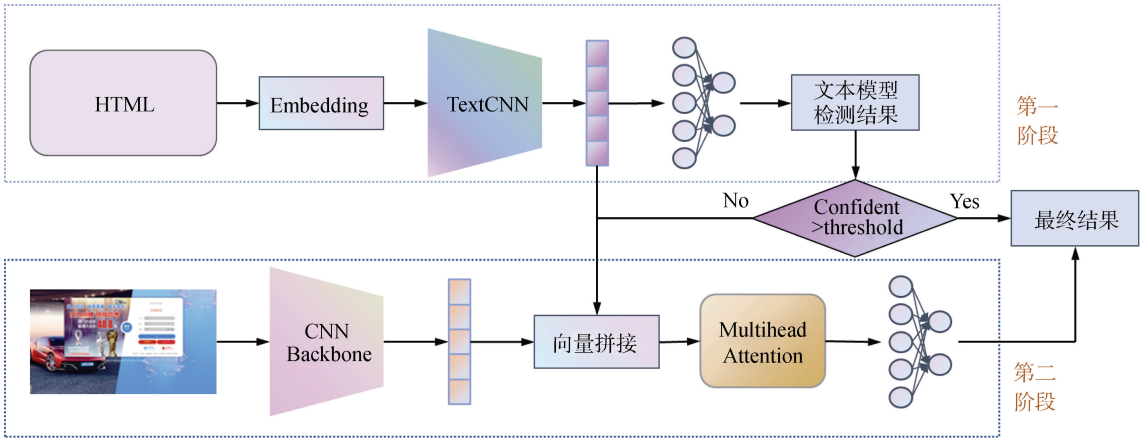

图 7 两阶段黑产网站识别框架

Figure 7 Two-Stage Illicit website identification framework

我们的模型由于设置了阈值，导致训练过程比较复杂。训练算法如算法 1 所示。在训练过程中采用前 1/2 轮只训练文本模型，将图像模型参数冻结，后 1/2 轮则相反。这样设置训练算法是由于第二阶段模型的识别需要一个优秀的文本表征向量，所以需要先将文本模型参数训练好才能训练图像模型参数。而在第二阶段训练中由于没有前向运算文本模型的 MLP 分类头，因此如果在训练第二阶段图像模型时不冻结文本模型参数，会导致文本模型参数变动使得文本分类效果变差。因此训练过程并没有考虑阈值的影响。

算法 1. 模型训练算法.

输入: 数据集 $D=\{(t_1,i_1,y_1),(t_2,i_2,y_2),\cdots,(t_N,i_N,y_N)\}$，训练轮数 E

输出: 模型 *model*

```
初始化模型 model
dev_best_loss = float('inf')
FOR epoch in [1, 2,⋯,E] DO
    IF epoch < ⌊E/2⌋ THEN
        FOR layerName,layer in model.layers DO
            IF 'img' in layerName THEN
                layer.trainable = False
            ELSE
                layer.trainable = True
            END IF
        END FOR
    ELSE
        FOR layerName,layer in model.layers DO
            IF 'text' in layerName THEN
                layer.trainable = False
            ELSE
                layer.trainable = True
            END IF
        END FOR
    END IF
    IF epoch == ⌊E/2⌋ THEN
        dev_best_loss = float('inf')
    END IF
    Train model with D
END FOR
RETURN model
```

在模型预测阶段，则将阈值加入进来。预测算法如算法 2 所示。根据文本模型的结果决定是否进入

第二阶段的图像判别。

算法 2. 模型预测算法.

输入: 数据集 $D=\{(t_1,i_1),(t_2,i_2),\cdots,(t_N,i_N)\}$，进入第二阶段阈值 *Threshold*

输出: 样本类别 *cls*

```
加载模型 model
FOR n in [1,2,…N] DO
    tRes,textFeature = model(t_n,'text')
    IF tRes.confident > Threshold THEN
        cls = tRes.cls
        RETURN cls
    ELSE
        cls = model(i_n,'img',textFeature)
        RETURN cls
    END IF
END FOR
```

使用该模型主要是为了在后面域名变换生成模型生成域名后, 对其进行判别, 找到其中潜藏的黑产网站。

3.3 多角度的训练数据构建

好的训练数据对于训练一个有效的域名变换生成模型至关重要, 因此需要采取有效的策略去构建数据集。通过对大量黑产域名的观察, 我们发现黑产域名存在着结构相似性, 这种具有部分相似的域名对比较适合模型学习。但是如果采用遍历所有可能的域名对来寻找这种具有结构相似性的域名对, 会导致冗余计算过多, 效率低下。因此我们采用了表征聚类方法缩小域名对的个数。在第 3.1 节中将我们收集的黑产网站域名进行聚类, 之后在得到的聚类结果中, 将相同类别的域名遍历构造域名对, 之后进行相似度的筛选来获得训练数据集。

对于 SSL 证书生成训练数据时和聚类类似, 我们取出黑产网站的 SSL 证书中的“主题替代名称”字段, 构成一个域名列表, 遍历每个域名列表构造域名对, 之后进行相似度的筛选获的训练数据集。这两种域名构造方法如图 8 所示。

在相似度筛选过程中采用的是 python 库中的 difflib.SequenceMatcher 方法。该方法的计算步骤是: 首先, SequenceMatcher 将输入的两个域名转换为字符序列。然后, 它找到这两个字符序列之间的最长公共子序列(LCS)。最后, 它根据 LCS 的长度以及输入域名的长度来计算相似度。相似度的计算公式为 (2×*LCS* 长度)/(域名 1 长度+域名 2 长度)。这种相似度计算方式更适合寻找存在部分结构相似的域名, 使得 Transformer 模型更易收敛。

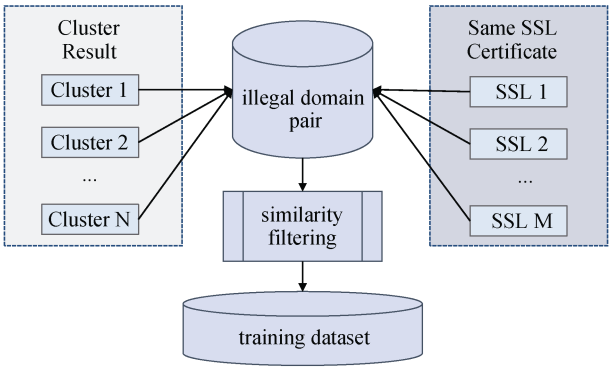

图 8 多角度训练数据构造方法

Figure 8 Multi-Perspective training data construction method

3.4 基于 Transformer 的域名变换方法

2017 年, Vaswani 等提出了 Transformer[40]模型并将其应用到了机器翻译领域。Transformer 模型完全基于注意力机制实现, 放弃了以往的卷积结构和递归结构, 如图 9 所示。Transformer 采用编码器及解码器架构, 并采用多头注意力机制来提高对特征的提取能力, 编码器捕获数据的隐藏特征, 并将该隐藏特征传给解码器后获得输出序列。

在本篇论文中未对 Transformer 模型结构做出更改, 因此对于每个模块的详细介绍可以参见文献[40]。

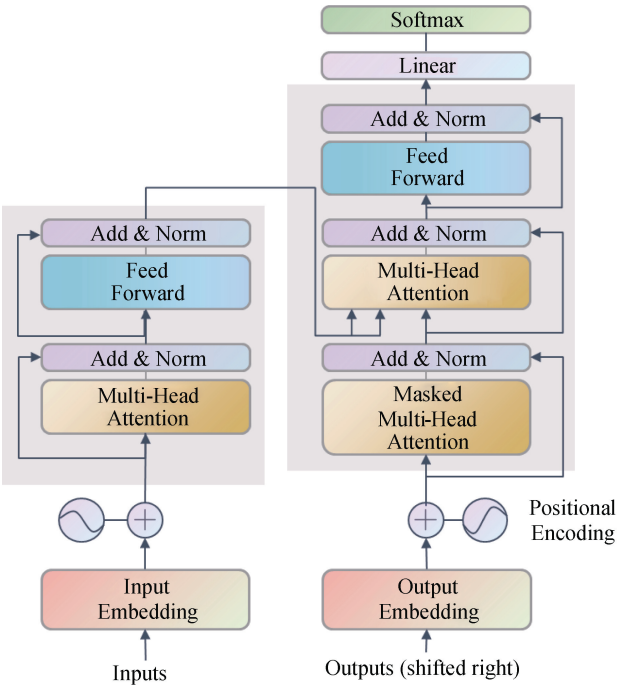

图 9 Transformer 模型结构

Figure 9 Transformer model architecture

3.4.1 训练阶段

在图 9 中的“Multi-Head Attention”模块中, 我

们设置 head 数 h=8。基于 Transformer 模型，我们提出了一种域名变换生成方法，该方法名字是 DNTrans。假设已知的黑产域名为 $dn_{illegal}$，那么要生成的疑似黑产域名就为 $dn_{generate}$。那么我们的目标就可以转化为对条件概率 $P(dn_{generate}\,|\,dn_{illegal})$ 建模，如公式(1)所示:

$$P(dn_{generate}\,|\,dn_{illegal}) = P(y_1,\cdots,y_m\,|\,x_1,\cdots,x_n) = \prod_{i=1}^{m} P(y_i\,|\,x_1,\cdots,x_n,y_1,\cdots,y_{i-1}) \tag{1}$$

域名和自然语言不同，域名更关注字符级别的特征。并且域名都是由字母 a-z、数字 0-9 和字符“.-”组成，因此我们的字典集合就由这些字符组成。这里需要注意的是，域名中的顶级域通常是由几个字母组成的不可分割的整体，但是为了简化模型的处理，我们没有将其单独提取出来作为一个整体，而是和二级域名一样的处理方式，切分为单个字符进行处理。通过实验证明这种方式也是有效的。

如图 10 所示，DNTrans 模型由编码器和解码器组成。首先我们将黑产域名输入编码器中，得到编码器的输出 e_{out}。然后，可以将 e_{out} 和起始字符 σ 输入到解码器中得到 y_1 和 $P(y_1\,|\,x_1,\cdots,x_n)$。以此类推，我们将 y_1 和 e_{out} 输入到解码器中可以得到 y_2 和 $P(y_2\,|\,x_1,\cdots,x_n,y_1)$。最后解码器输出为结束字符 ω 时生成终止，这时可以得到 $dn_{generate}$ 和 $P(dn_{generate}\,|\,dn_{illegal})$。此时可以通过生成的域名计算损失后回传损失来优化模型参数。

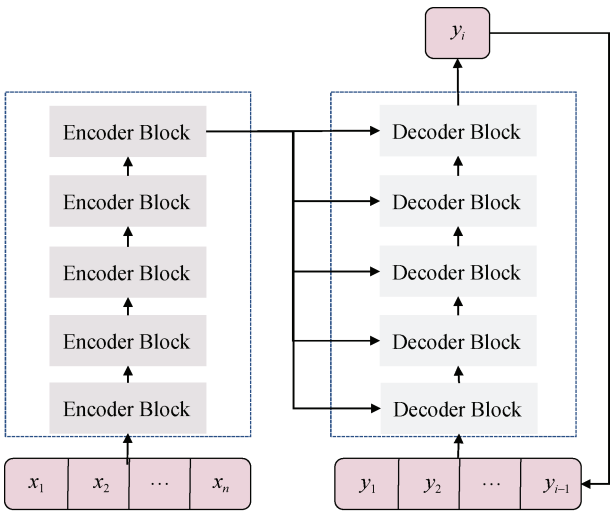

图 10　通过 DNTrans 生成 y_i

Figure 10　Generate y_i through DNTrans

在我们模型中，编码器和解码器的层数都为 5 层。多头注意力模块中 head 的数量设置为 8，维度为 512。前馈神经网络模块神经元个数为 1024。

3.4.2　生成阶段

为了产生多个潜在黑产域名，DNTrans 采用了基于队列的搜索策略，如图 11 所示。对于域名的生成从起始字符 σ 开始，每次生成时会将概率 $P(dn_{prefix}\,|\,dn_{illegal})$ 求出来，如公式(2)所示。

$$P(dn_{prefix}\,|\,dn_{illegal}) = P(y_1,\cdots,y_p\,|\,x_1,\cdots,x_n) = \prod_{i=1}^{p} P(y_p\,|\,x_1,\cdots,x_n,y_1,\cdots,y_{p-1}) \tag{2}$$

其中，dn_{prefix} 是已生成的部分前缀，每次生成前缀后，如果该前缀的概率 $P(dn_{prefix}\,|\,dn_{illegal})$ 大于我们设置的概率阈值(实验中设置为 0.00001)时，会将其加入到前缀队列中，等待后续继续生成。在生成过程中，如果遇到结束字符 ω 并且其概率值大于阈值，就将其输出为一个变换的域名。通过循环判断前缀队列是否为空，不为空时就取队头元素继续生成，如果为空则表示生成结束。

如图 11 所示是该生成算法的一个示例，输入起始字符 σ 后判断每个字符的概率是否超过阈值，图中假设 a,z,9 超过了阈值，就将 a,z,9 加入前缀队列，之后取队列队头元素 a 继续生成，假设 b,0 超过阈值，此时就将 ab,a0 加入前缀队列。以此类推，直到队列为空。

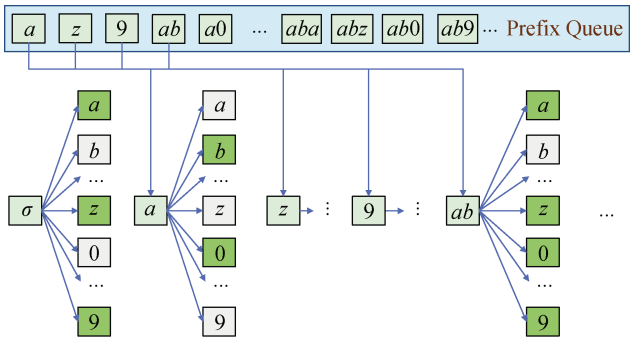

图 11　域名生成算法

Figure 11　Domain name generation algorithm

4　实验数据与结果分析

在这个章节中，我们进行了实验来评估所提出的黑产域名变换生成方案。我们的实验环境是一台配备了 Intel(R) Xeon(R) Gold 6230R CPU,256GB 内存,Nvidia Geforce RTX4090,24GB 显存的工作站。

数据集。我们的数据集分为两种，第一种是为了训练两阶段黑产网站检测模型用到的数据。第一种

数据包括两类网站, 一类是正常网站, 一类是黑产网站。黑产网站来自于之前工作发现的黑产网站[42]。正常网站是从互联网上通过爬虫抓取获得的。黑产网站和正常网站个数总和为 36124 个, 其中黑产网站 18000 个, 正常网站 18124 个。第二种是为了训练域名变换生成方法的黑产域名数据集, 我们收集了将近 28 万条黑产网站的域名。

评价指标。在本文中, 我们使用了准确率 *Accuracy(ACC)*, 精确率 *Precision*, 召回率 *Recall*, F1 分数和执行时间来评估黑产网站识别模型的性能。具体公式如下:

$$\begin{aligned} ACC &= \frac{TP+TN}{TP+FP+TN+FN} \\ Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \end{aligned} \tag{3}$$

其中, *TP* 是真正例, *TN* 是真负例, *FP* 是假正例, *FN* 是假负例。

F1-*score* 可以通过 *Precision* 和 *Recall* 进行计算:

$$\mathrm{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

执行时间则是在相同条件下, 采用单进程单线程在 60000 个样本上进行模型预测所需花费的时间。

采用生成域名可访问比率 *DAR* 和黑产域名比率 *IDR* 来衡量域名变换生成的性能, 计算公式如下:

$$\begin{aligned} DAR &= \frac{AD}{GD} \\ IDR &= \frac{ID}{AD} \end{aligned} \tag{5}$$

其中, *AD* 表示可访问域名的个数, *GD* 表示生成的域名个数, *ID* 表示可访问域名中黑产域名个数。

4.1 域名表征聚类实验结果

域名表征采用的是预测下一字符的代理任务进行无监督学习的。我们尝试了两种方法, 第一种是 LSTM, 第二种是双向 LSTM(Bi-LSTM)。这两种方法各训练 15 轮, 得到如图 12 所示的模型损失变化图。从图 12 中可以看出来 Bi-LSTM 表现的更好更稳定, 而 LSTM 波动较大, 模型参数也无法收敛。这个现象可能是由于这个代理任务的选择原因, 因为 Bi-LSTM 能够看到后面的字符, 因此其对字符的预测更准确。由于黑产域名存在较大的随机性, LSTM 只能看到前面的字符, 这个预测任务对于 LSTM 来说过于困难了。

因此, 通过 Bi-LSTM 和 LSTM 的 loss 值的变化, 我们最终选择了 Bi-LSTM 模型作为域名表征模型。我们设置表征向量的维度为 64, 即每个域名被转换为一个 64 维的向量。这个向量蕴含着域名的上下文特征, 对其进行聚类后可以构成域名变换训练数据集。

对于域名聚类, 我们尝试了多种算法, 首先使用了 DBSCAN 算法进行聚类。DBSCAN 算法是一种基于密度的算法, 该算法有两个超参数需要调整, 分别是邻域半径 ε 和成为核心对象的在领域半径内的最少点数(*minPts*)。我们尝试了多种参数组合, 发现总是出现两个问题, 第一异常点过多, 很多域名被分类为-1 类, 即无法分类到某个具体类别的域名; 第二某些类别中的域名过多, 这种结果会导致无法有效的降低构造域名变换训练数据集的复杂度。因此我们认为 DBSCAN 算法不适合我们这个任务。

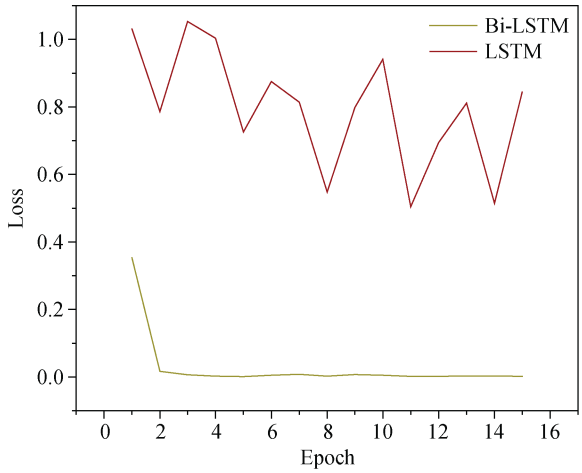

图 12 域名表征训练 loss

Figure 12 Domain name representation training loss

我们主要测试了其余两个聚类算法, 一种是传统的机器学习算法层次聚类算法, 另一种是深度聚类算法 SCAN 算法[41]。层次聚类算法既支持设定类别数进行聚类, 也可以不设定类别数进行聚类。本文采用不设定类别数的方法, 此时层次聚类只有一个参数需要调整, 即距离阈值参数。深度聚类算法 SCAN 是一种需要设定类别数进行聚类的算法, 我们尝试了多个不同的类别数进行实验。

我们采用了两个聚类常用的评价指标来衡量两种算法的聚类性能, 分别为轮廓系数和 DBI (Davies-Bouldin Index)。轮廓系数的取值范围在–1 到 1 之间。取值接近于 1 表示样本与自身的聚类更紧密, 与其他聚类之间更分离, 表示聚类结果较好。取值接近于–1 表示样本与自身的聚类较差, 与其他聚类之间重叠较多, 表示聚类结果较差。DBI 的取值范围是非负实数。最小值为 0, 表示聚类结果的紧密度和分

离度最佳。较大的 DBI 值表示聚类结果的紧密度较差或分离度较差。因此, DBI 越接近于 0, 表示聚类结果的质量越好。

两种聚类算法的结果如图 13 所示。其中图 13a 为 SCAN 聚类算法的实验结果, 横坐标为不同的聚类类别数, 纵坐标左轴为轮廓系数的值, 右轴为 DBI 的值。图 13b 为层次聚类算法的实验结果, 横坐标为不同的距离阈值, 纵坐标和 SCAN 算法的纵坐标一致。

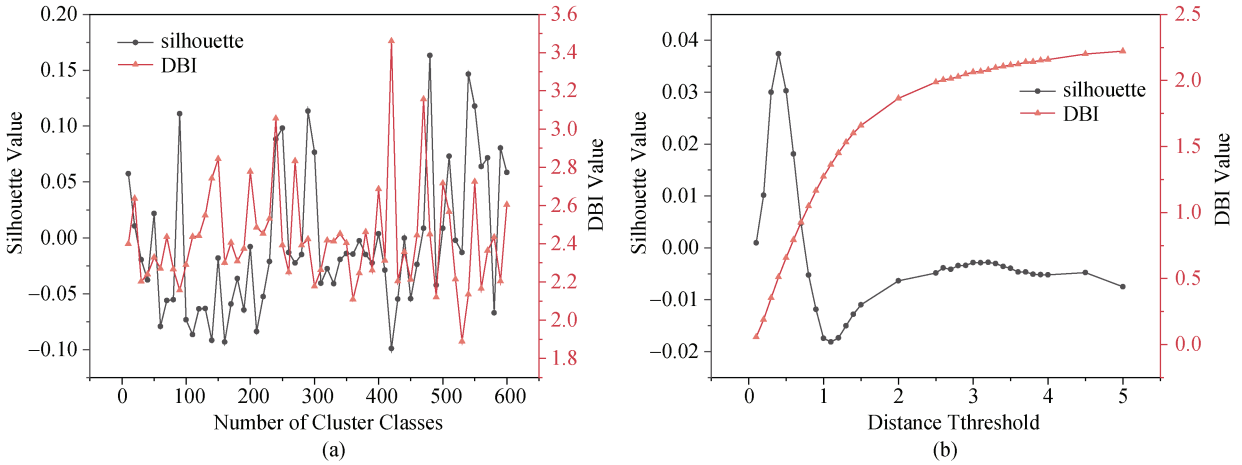

图 13 聚类算法性能指标 (a)SCAN 聚类评价指标; (b)层次聚类评价指标

Figure 13 Cluster algorithm performance metrics: (a) SCAN Clustering Evaluation Metrics; (b) Hierarchical clustering evaluation metrics

SCAN 聚类算法由于必须指定类别数, 需要确定类别数, 而图 13a 中评价指标的变化没有规律, 并且波动很大, 因此难以确定合适的类别数。图 13b 中层次聚类算法的指标随着距离阈值变化可以明显看出规律。因此我们最终选择了层次聚类算法作为域名聚类算法, 并且综合两个评价指标后取距离阈值为 0.4。为了有效提高构造数据集的效率, 我们需要保证聚类结果尽可能均匀并且每一类中的样本数保持适中, 既不能太多, 也不能太少。如图 14 所示, 横坐标表示聚类结果中每一类包含的样本数量, 纵坐标表示聚类簇的个数。从图 14 可以看出大部分聚类簇中包含的样本数集中在 6 个左右, 这个分布非常适合我们训练样本的构造。

4.2 两阶段黑产网站检测结果

我们首先在单独文本模型和图像模型上分别做了实验, 通过实验结果选择合适的文本模型和图像模型进行组合, 构成两阶段黑产网站检测模型。

首先是文本模型选择, 我们在文本数据上训练了 5 种算法模型进行对比。RNN、GRU 和 LSTM 模型层数都设置为 2, 隐藏层大小都设置为 128。Bi-LSTM 相比于前面三个模型, 模型层数和隐藏层大小都一样, 只是设置了双向。在网页 HTML 数据上实验时设置句子最大长度为 256, HTML 文本数据模型评估结果如表 1 所示。表中的 Predict-time-6000 这一列数据是采用单进程单线程模式识别 60000 个样本所需的时间, “s”表示秒。从表中的实验结果可以明显观察到 TextCNN 模型具有最优的表现, 不仅准确率最高, 而且其预测速度也最快, 因此我们选择了 TextCNN 作为两阶段黑产网站识别模型中的文本端模型。

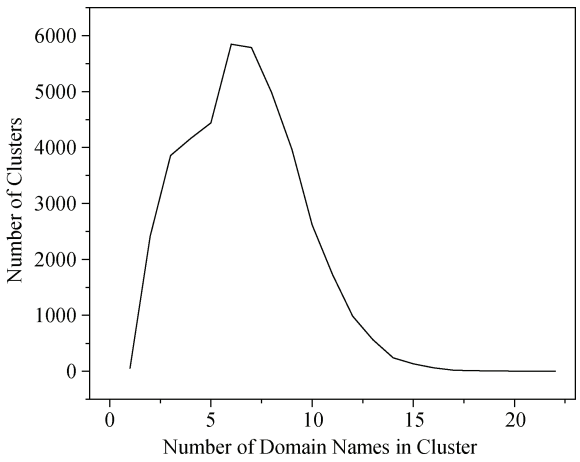

图 14 聚类结果的每一类中包含的样本数量分布

Figure 14 Distribution of sample counts in each cluster class

其次是图像模型选择, 我们在图像上也训练了 5 个模型进行黑产网站的识别。为了对比不同卷积神经网络的性能, 我们自己也构建了两个 CNN 模型, 其命名为 CNN-7 和 CNN-10。CNN-7 是由 4 个卷积层、3 个全连接层构成。CNN-10 是由 5 个卷积层和

表 1 HTML 模型评估结果

Table 1 HTML model evaluation results

Model	ACC	Precision	Recall	F1	Predict-time-60000
RNN	92.33	93.28	92.19	92.27	65s
GRU	96.37	96.37	96.39	96.37	70s
LSTM	96.03	96.03	96.04	96.03	103s
Bi-LSTM	97.53	97.53	97.54	97.53	164s
TextCNN	**98.33**	**98.33**	**98.33**	**98.33**	**64s**

5 个全连接层构成。其余三个 CNN 模型是 ResNet 系列模型, 分别为 ResNet18、ResNet34 和 ResNet50。这些 ResNet 系列模型最后都添加了 MLP 分类头, 以使其适合我们的分类任务。图像分类模型评估结果如表 2 所示。表 2 中 Predict-time-6000 中, 由于图像模型在从硬盘获取数据时需要的时间较长, 因此这里“m”表示分钟, “s”表示秒。在表中可以观察到随着模型层数的加深, 模型参数的增多, 模型的识别效果也在提升。但是也可以看到不同的模型识别速度也不一样, 模型深度小, 模型参数少的模型速度较快。识别准确率最高的模型 ResNet50 也是识别 60000 个样本所需时间最长的模型。这个时间相比于在文本上表现最好的 TextCNN 模型高了将近 50 倍。这个时间花费在大规模数据集上是难以忍受的。在最终模型选择上, 我们选择了各种表现都较优的 ResNet34 模型。

最终我们将文本模型和图像模型的分类头去掉, 保留模型中的特征提取部分后组后为两阶段模型。两阶段融合模型评估中, 我们做了 5 种不同置信度参数的实验。这个置信度可以控制一个样本是否需要加入图像特征来进行识别。HTML 文本和图像数据实验结果如表 3 所示。

表 2 图像模型评估结果

Table 2 Image model evaluation results

Model	ACC	Precision	Recall	F1	Predict-time-60000
CNN-7	98.37	98.4	98.35	98.37	**31m48s(1908s)**
CNN-10	98.73	98.75	98.72	98.73	33m22s(2002s)
ResNet18	99.43	99.44	99.43	99.43	37m21s(2241s)
ResNet34	99.5	99.5	99.5	99.5	42m10s(2530s)
ResNet50	**99.57**	**99.57**	**99.57**	**99.57**	48m4s(2884s)

表 3 两阶段模型评估结果

Table 3 Two-stage model evaluation results

Model	ACC	Precision	Recall	F1	Predict-time-60000
confidence-0.8	99.63	99.64	99.63	99.63	**96s**
confidence-0.85	99.7	99.7	99.7	99.7	101s
confidence-0.9	99.73	99.74	99.73	99.73	125s
confidence-0.95	**99.77**	**99.77**	**99.76**	**99.77**	168s
confidence-0.99	99.77	99.77	99.76	99.77	321s

两阶段模型实验结果中 confidence 表示两阶段检测模型中文本模型检测结果的置信度小于某值时进入第二阶段进行图文融合检测的阈值。

综合表 1、表 2 和表 3 中可以看出只使用文本检测时效率时最高的, TextCNN 模型只需要 64 秒就可以在单进程单线程模式下检测完 60000 个样本, 属于效率最高的模型, 但是由于网站的伪装, 导致其准确率时所有模型中最低的。单独使用图像模型 ResNet34, 需要超过 42 分钟的时间才能检测完 60000 个样本, 尽管准确率相比 TextCNN 模型高, 但是效率太低无法使用。

两阶段检测模型可以设置不同的 confidence 阈值, 从表中的数据可以看到随着置信度阈值的提高, 准确率也在上升, 所花费的时间也在增加。这是由于置信度阈值控制着样本是否进入第二阶段的检测。如果样本在文本模型的识别中识别结果置信度高于

阈值, 那么就会直接输出结果来加速识别速度。反之, 如果文本模型的识别结果置信度低于所设置的阈值, 那么样本就会进入第二阶段, 此时会从磁盘中读取该样本对应的图像数据后传入图像模型获取其表征, 然后结合第一阶段中得到的文本表征一起传入注意力机制的融合模型中进行识别来提高准确率。通过表 3 中可以看到, 两阶段模型准确率最高达到了 99.77%, 并且识别效率相比于单独使用图像模型有着巨大的提高。

从上面的实验结果来看, 我们所提的多模态数据融合的两阶段黑产网站识别方法不仅在准确率上相比于单一模态要高, 而且对于识别效率的提升更加明显。我们在实验中的测试是单进程单线程方式进行, 在实际应用中, 我们可以使用多进程多线程的方式来进一步提高识别效率。在我们所提方法的帮助下, 完全可以实现对大规模网站样本的黑产网站识别。

4.3 生成新的黑产域名

通过第 3.3 节中两种方式一共生成了 1896068 对数据, 其中划分了训练集和测试集, 训练集 1516804 对, 测试集 379214 对。通过对每对数据计算相似度, 然后根据相似度阈值进行筛选, 我们设置了 6 个不同的阈值, 每种阈值筛选后保留的训练集数量如图 15 所示。可以看到相似度阈值在小于 0.4 时, 训练集数量变化并不明显。因此我们没有对阈值小于 0.4 的数据构建训练集。

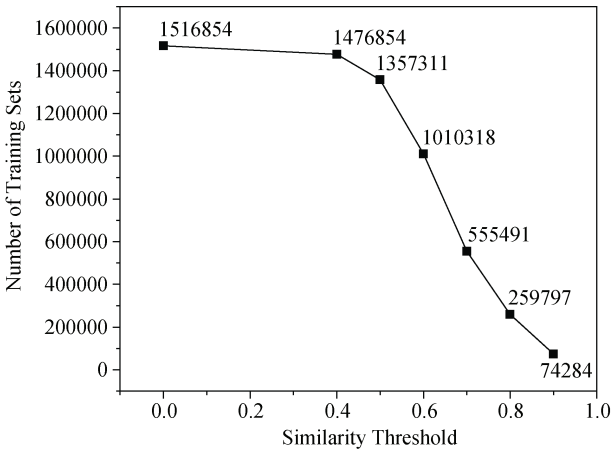

图 15 不同相似度阈值下的训练集数量变化

Figure 15 Variation in training set size at different similarity thresholds

针对每种相似度阈值构造的数据集, 我们分别训练了 6 个模型, 分别为 0.4、0.5、0.6、0.7、0.8 和 0.9, 每个模型参数一致。训练时的超参数设置也保持一致, 每个模型训练 20 轮。训练过程 loss 变化如图 16 所示。从图中可以看到同样的训练轮数, 相似度越高的样本训练出来的模型 loss 值越低。这是因为相似度越高的样本, 其训练时的输入域名和目标输出域名重合度越高, 这对于模型来说任务变得简单了。因此我们无法单独只从 loss 值来判断选择某个数据集训练得到的模型。

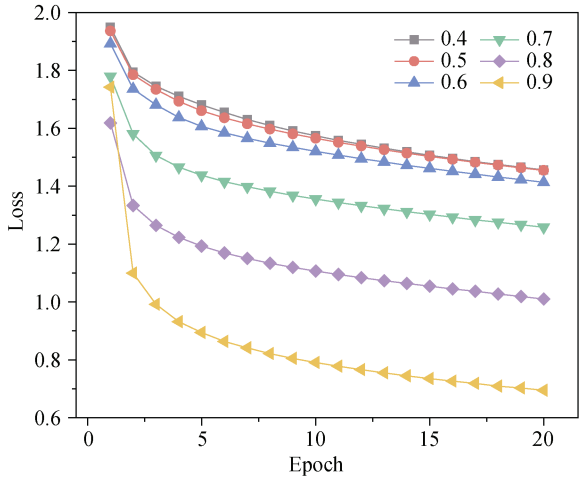

图 16 不同相似度阈值下的训练 loss 变化

Figure 16 Change in training loss at different similarity thresholds

为了对比出生成能力更好的模型, 我们在 20 个黑产域名数据上做变换生成, 其中有 10 个域名包含数字, 其余 10 个域名不包含数字。为了能高效的选择合适的模型, 我们对这 20 个域名变换生成的域名进行分析。我们评估了每个域名生成域名的数量, 存活性检测后存活域名数量, 可成功抓取网页的域名数量, 黑产网站域名数量等。域名存活性检测采用 socket 连接测试进行, 不进行 HTTP 请求获取其网页内容, 速度相对较快。

如图 17 上半部分为域名生成数量, 可以看到相似度筛选阈值为 0.4、0.5 训练得到的模型生成数量较少, 每个域名变换生成的数量普遍在 2000 个以内。阈值为 0.6、0.7、0.8 和 0.9 的模型生成数量波动较大, 在某些黑产域名作为输入时生成较多, 某些较少。图 17 下半部分表示每个域名生成的域名通过 socket 存活性检测后存活的域名数量。从图中可以看到阈值为 0.4、0.5 和 0.9 的域名生成的域名存活数量较少, 0.4 和 0.5 在生成域名时就生成的比较少, 0.9 属于生成的较多, 但是存活较少, 在这个存活性检测指标表现上比较差。0.6、0.7 和 0.8 的表现在这个图中难以区分优劣。图 18 中展示了每个域名生成的域名中存活域名所占的比例。可以在图中看到阈值为 0.6 时存活域名比例较高, 而图中重点显示的阈

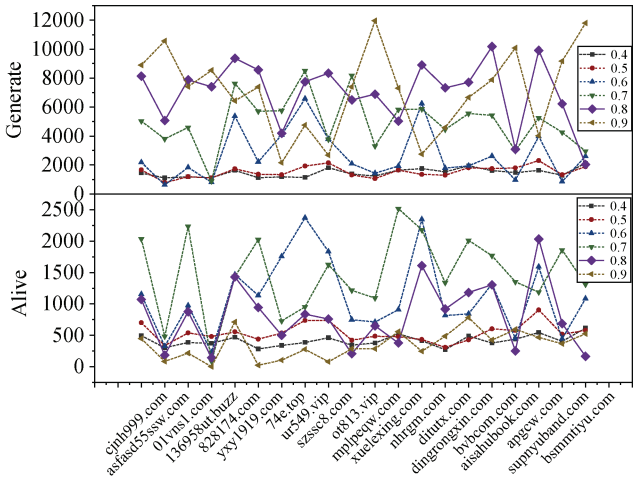

图 17　6 个不同相似度数据训练后的模型变换生成域名数量及存活数量对比

Figure 17　Comparison of the number of transformed generated domain names and surviving domain names after training with 6 different similarity datasets

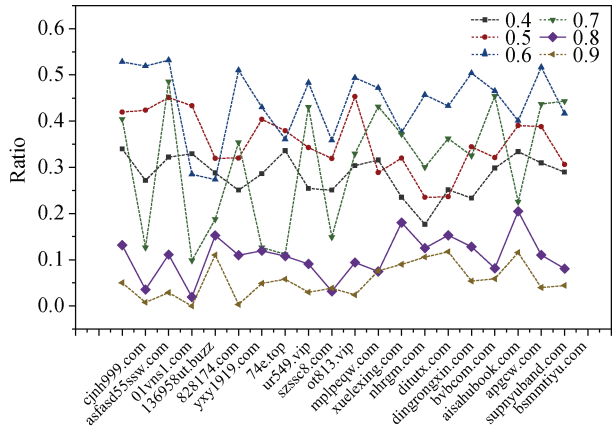

图 18　6 个不同相似度数据训练后的模型变换生成域名数量中存活域名数量所占比例

Figure 18　Proportion of surviving domain names in the number of transformed generated domain names after training with 6 different similarity datasets

值为 0.8 的这条线在这个评价指标上并不是最优的。至于将 0.8 突出显示的原因是在后面综合黑产网站个数以及比例评价指标中，阈值设置 0.8 被认为是最合适该任务的。

图 19 为 6 个模型生成的域名中可成功爬取网页的数量以及为黑产网站的数量。其中上半部分为可成功爬取的域名数量。我们将前面通过 socket 存活性检测的域名采用进行抓取，首先使用 HTTPS 协议进行，如果采用 HTTPS 协议访问失败，再采用 HTTP 协议访问，如果两次访问都失败，就将其标志为不可抓取。可成功抓取的数量相比于 socket 检测存活的数量少了很多，通过观察发现有一部分网站被网络运营商封锁导致无法访问，也有一部分网站已经关停导致无法访问。通过图 19 上半部分可以看到可爬取的域名数量中，阈值设置为 0.7 的模型表现最好，阈值设置为 0.6 和 0.8 时模型表现次之。阈值设置为 0、4、0.5 和 0.9 时，每个域名可爬取的数量变化不剧烈，数量也都较少。这种情况表明了，相似度阈值设置的过低或者过高都会导致模型效果变差。图 19 下半部分展示了每个域名变换生成的黑产域名的数量。图中横坐标为 20 个黑产域名，前 10 个域名包含数字，后 10 个域名则不包含。可以看到所有的模型对包含数字的域名变换生成效果相比于不包含数字的域名要好。其中阈值设置为 0.7 和 0.8 时黑产域名变换生成效果比较好。可以看到生成黑产域名数量最高的是 828174.com，从该域名最多扩展出了 207 个黑产域名。

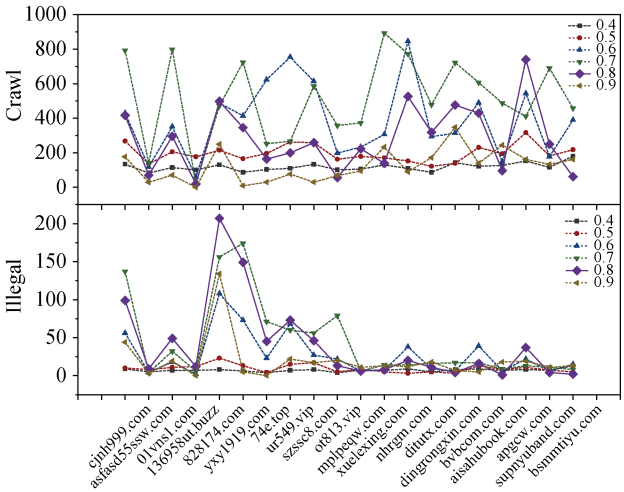

图 19　6 个不同相似度数据训练后的模型变换生成域名可成功爬取网页数量及黑产网站数量对比

Figure 19　Comparison of successfully crawled web-page count and illicit website count of transformed generated domain names by the model trained with 6 different similarity datasets

图 20 是将每个模型对 20 个黑产域名变换生成的可抓取到的域名总数，每个模型生成的黑产域名数量以及黑产域名占可抓取域名总数的比例。所有模型总共生成了 32505 个可抓取的域名。其中阈值设置为 0.7 时变换生成可抓取的域名数量最多，达到了 10303 个，并且其中黑产域名的数量也是最多的，达到了 900 个。但是阈值为 0.7 时黑产域名占可抓取域名的比例却不高。黑产域名占可抓取域名的比例最高的时阈值设置为 0.9，但是阈值设置为 0.9 时生成的新黑产域名数量却不多。相比之下，阈值设置为 0.8 时生成的新黑产域名数量和比例都较高。新黑产

域名生成的数量以及比例是最重要的参考指标，并且结合前面的评价指标综合考虑后，我们选择了阈值设置为 0.8 的模型。其生成的黑产域名的占所有可爬取域名的 14.54%。通过 20 个域名变换生成了 811 个黑产域名，扩展倍数达到了 40.55 倍。如果输入的域名全为包含数字的域名，其扩展比例将会更高。

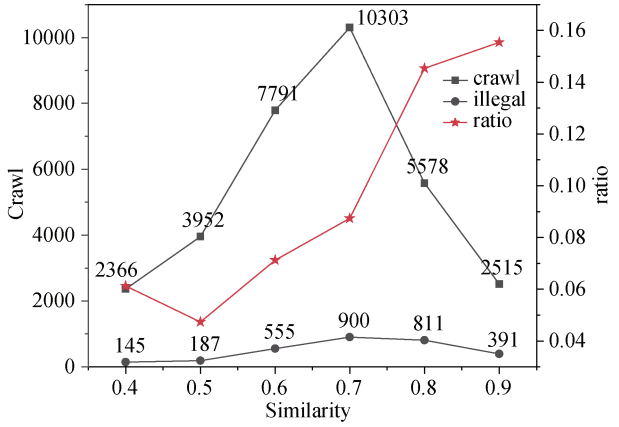

图 20　6 个模型变换生成域名可成功爬取网页总数量及黑产网站数量以及比例

Figure 20　Total count of successfully crawled webpages and illicit website count, along with the proportion, for transformed generated domain names by 6 models

为了更好的评估表现最好模型的性能，我们选择了 100 个黑产网站域名使用阈值为 0.8 训练的得到的域名变换生成模型进行生成。作为模型输入的 100 个域名，其中 70 个为含有数字的域名，30 个不含数字的域名。通过这 100 个域名，我们一共生成了 1383705 个域名，其中 229270 个可使用 socket 模块的 gethostbyname 函数获取 IP 地址的域名。通过 HTTP 或 HTTPS 请求访问域名后得到 188463 个可抓取网页域名。我们采用两阶段黑产网站检测模型对其进行检测，发现了 35998 个黑产域名。因此可以计算出生成域名可访问比率 DAR=(188463/1383705)=13.62%，以及可访问域名中黑产域名比率 IDR=(35998/188463)=19.1%，黑产域名的扩展倍数达到了 359.98，即通过一个黑产域名可以平均扩展出 360 个新的黑产域名。

文献[34]中使用 GAN 进行赌博、色情域名的生成，我们的实验结果与其对比如表 4 所示。本文方法主要面向对某个黑产域名的扩展，因此在黑产域名比率上略低于文献[34]中的 GAN 方法，但是在扩展倍数这一指标上，本文方法有着巨大的提升。

我们针对这 100 个域名是否含有数字字符分为了两类进行分析。通过 70 个包含数字的黑产域名变换生成了 34464 个新的黑产域名，平均每个域名生成了 492 个黑产域名。通过 30 个不包含数字的黑产域名变换生成了 1534 个新的黑产域名，平均每个域名生成了 51 个黑产域名。这个 DNTrans 域名变换生成模型更适合对域名中包含数字的域名进行扩展，这也符合第 2.1.1 中发现的黑产域名分布特征。

阈值设置。在本文实验中共有四个阈值需要设置，分别是层次聚类距离阈值、两阶段模型置信度阈值、数据集构建相似度阈值和域名生成概率阈值。我们对不同阈值进行实验，最终得到如表 5 所示的最佳阈值配置。

表 4　黑产域名生成结果

Table 4　Illegal domain name generation results

生成方法	黑产域名比率	扩展倍数
DNTrans(本文)	19.1	359.98
文献[34]	23.82	7.5

表 5　阈值配置表

Table 5　Threshold Configuration

阈值名称	阈值数值
层次聚类距离阈值	0.4
两阶段模型置信度阈值	0.95
数据集构建相似度阈值	0.8
域名生成概率阈值	0.00001

5　总结

本文介绍了一种基于 Transformer 的域名变换生成模型 DNTrans，用于对已有黑产域名进行变换生成，扩展出更多相关未知的黑产域名。DNTrans 使用 Transformer 神经网络结构，可以在每次生成域名时有已知黑产域名作为辅助信息，避免了通过固定路径生成域名。最终，实验结果证明了 DNTrans 在进行域名扩展方面的有效性，平均一个已知黑产域名可以扩展出 360 个新的黑产域名。

尽管所提的方法能够有效的扩展已知黑产域名，但是本方法还存在一些局限性。我们针对每个域名进行单独生成，根据生成时的算法，会比较倾向于生成长度较短的域名，未来可以考虑增加长度归一化或者一些惩罚项来改进。

参考文献

[1] Prakash P, Kumar M, Kompella R R, et al. PhishNet: Predictive Blacklisting to Detect Phishing Attacks[C]. *2010 Proceedings IEEE INFOCOM*, 2010: 1-5.

[2] Akiyama M, Yagi T, Itoh M. Searching Structural Neighborhood of

Malicious URLs to Improve Blacklisting[C]. *2011 IEEE/IPSJ International Symposium on Applications and the Internet*, 2011: 1-10.

[3] Rao R S, Pais A R. An Enhanced Blacklist Method to Detect Phishing Websites[C]. *Information Systems Security*, 2017: 323-333.

[4] Liu W Y, Liu G, Qiu B T, et al. Antiphishing through Phishing Target Discovery[J]. *IEEE Internet Computing*, 2012, 16(2): 52-61.

[5] Zou F T, Gang Y X, Pei B, et al. Web Phishing Detection Based on Graph Mining[C]. *2016 2nd IEEE International Conference on Computer and Communications*, 2016: 1061-1066.

[6] Zhao J, Shao M L, Peng H, et al. Porn2Vec: A Robust Framework for Detecting Pornographic Websites Based on Contrastive Learning[J]. *Knowledge-Based Systems*, 2021, 228: 107296.

[7] Li Y F, Yu L J, Liu Q Y. HinPage: Illegal and Harmful Webpage Identification Using Transductive Classification[C]. *Information Security and Cryptology*, 2023: 373-390.

[8] Ma J, Saul L K, Savage S, et al. Beyond Blacklists[C]. *The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009: 1245-1254.

[9] Yuan H P, Yang Z G, Chen X, et al. URL2Vec: URL Modeling with Character Embeddings for Fast and Accurate Phishing Website Detection[C]. *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications*, *Ubiquitous Computing & Communications*, *Big Data & Cloud Computing*, *Social Computing & Networking*, *Sustainable Computing & Communications*, 2018: 265-272.

[10] Zhu E Z, Ju Y Y, Chen Z L, et al. DTOF-ANN: An Artificial Neural Network Phishing Detection Model Based on Decision Tree and Optimal Features[J]. *Applied Soft Computing*, 2020, 95: 106505.

[11] Yan X D, Xu Y, Cui B J, et al. Learning URL Embedding for Malicious Website Detection[J]. *IEEE Transactions on Industrial Informatics*, 2020, 16(10): 6673-6681.

[12] Yang R D, Zheng K F, Wu B, et al. Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning[J]. *Sensors*, 2021, 21(24): 8281.

[13] Sun G Y, Ye F, Chai T T, et al. Gambling Domain Name Recognition *via* Certificate and Textual Analysis[J]. *The Computer Journal*, 2023, 66(8): 1829-1839.

[14] Su Y, Peng B T, Li X D. Fast Illegal Webpage Detection Algorithm Based on Massive Domain Name Resolution Records[C]. *2022 IEEE International Conference on Big Data*, 2022: 4313-4322.

[15] Min M, Lee J J, Lee K. Detecting Illegal Online Gambling (IOG) Services in the Mobile Environment[J]. *Security and Communication Networks*, 2022, 2022(1): 3286623.

[16] Li L X, Gou G P, Xiong G, et al. Identifying Gambling and Porn Websites with Image Recognition[C]. *Advances in Multimedia Information Processing - PCM 2017*, 2018: 488-497.

[17] Liu D J, Lee J H, Wang W, et al. Malicious Websites Detection *via* CNN Based Screenshot Recognition[C]. *2019 International Conference on Intelligent Computing and its Emerging Applications*, 2019: 115-119.

[18] Jain A K, Gupta B B. A Machine Learning Based Approach for Phishing Detection Using Hyperlinks Information[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2019, 10(5): 2015-2028.

[19] Cernica I, Popescu N. Computer Vision Based Framework for Detecting Phishing Webpages[C]. *2020 19th RoEduNet Conference*: *Networking in Education and Research*, 2020: 1-4.

[20] Zhang W, Jiang Q S, Chen L F, et al. Two-Stage ELM for Phishing Web Pages Detection Using Hybrid Features[J]. *World Wide Web*, 2017, 20(4): 797-813.

[21] Zhu E Z, Chen Y Y, Ye C C, et al. OFS-NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network[J]. *IEEE Access*, 2019, 7: 73271-73284.

[22] Yang P, Zhao G Z, Zeng P. Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning[J]. *IEEE Access*, 2019, 7: 15196-15209.

[23] Chen Y, Zheng R F, Zhou A M, et al. Automatic Detection of Pornographic and Gambling Websites Based on Visual and Textual Content Using a Decision Mechanism[J]. *Sensors*, 2020, 20(14): 3989.

[24] Gandotra E, Gupta D. Improving Spoofed Website Detection Using Machine Learning[J]. *Cybernetics and Systems*, 2021, 52(2): 169-190.

[25] Xiong J Y. Recognition of Illegal Websites Based on Similarity of Sensitive Features of Mixed Elements[C]. *2022 International Conference on Computation*, *Big-Data and Engineering*, 2022: 9-12.

[26] Liu D J, Geng G G, Zhang X C. Multi-Scale Semantic Deep Fusion Models for Phishing Website Detection[J]. *Expert Systems with Applications*, 2022, 209: 118305.

[27] Bamital's DGA. Trojan bamital 13 en. https://docs.broadcom.com /doc/trojan-bamital-13-en. May. 2012.

[28] Dyre's DGA. Threat Spotlight: Dyre/Dyreza. https://blogs.cisco.com/security/talos/threat-spotlight-dyre. Mar. 2015.

[29] Conficker's DGA. Conficker Summary and Review. https://www.icann.org/en/system/files/files/conficker-summary-review-07may10-en.pdf. May. 2010.

[30] Pushdo's DGA. Pushdo Analysis. https://www.scribd.com/ document/514439673/THREAT-ANALYSIS-Pushdo. Dec. 2007.

[31] Matsnu's DGA. Matsnu malwareid technical brief. https://blog.check point.com/wp-content/uploads/2015/07/matsnu-malwareid-technical-brief.pdf. May. 2015.

[32] Suppobox's DGA. SUPPOBOX. https://know.netenrich.com/ threatin tel/malware/Suppobox. Mar. 2018.

[33] Gong B, Ning Z H, Zhu Y, et al. Character-Level Domain Name Generation Algorithm Based on ED-GAN[C]. *2022 11th International Conference on Software and Computer Applications*, 2022: 198-205.

[34] Liang Y C, Cheng Y N, Zhang Z X, et al. Illegal Domain Name Generation Algorithm Based on Character Similarity of Domain Name Structure[J]. *Applied Sciences*, 2023, 13(6): 4061.

[35] Pham T D, Pham T T T, Ta V C. En-SeqGAN: An Efficient Sequence Generation Model forDeceiving URL Classifiers[C]. *Recent Challenges in Intelligent Information and Database Systems*, 2022: 477-489.

[36] Valentim R V, Drago I, Trevisan M, et al. URLGEN-Toward Automatic URL Generation Using GANs[J]. *IEEE Transactions on*

Network and Service Management, 2023, 20(3): 3734-3746.

[37] Zhai Y, Yang J, Wang Z X, et al. Cdga: A GAN-Based Controllable Domain Generation Algorithm[C]. *2022 IEEE International Conference on Trust*, *Security and Privacy in Computing and Communications*, 2022: 352-360.

[38] Vecile S, Lacroix K, Grolinger K, et al. Malicious and Benign URL Dataset Generation Using Character-Level LSTM Models[C]. *2022 IEEE Conference on Dependable and Secure Computing*, 2022: 1-8.

[39] Wu C B, Fei J L. An Abnormal Domain Name Generation Method Based on a Character-Level Model[C]. *The 2022 4th International Conference on Robotics*, *Intelligent Control and Artificial Intelligence*, 2022: 804-810.

[40] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. *Advances in neural information processing systems*, 2017, 30.

[41] Van Gansbeke W, Vandenhende S, Georgoulis S, et al. SCAN: Learning to Classify Images without Labels[C]. Computer Vision – ECCV 2020. Cham: Springer International Publishing, 2020: 268-285.

[42] black website. kaggle. https://www.kaggle.com/datasets/listone/black-website. Mar. 2023.

王博 于 2021 年在新疆大学信息安全专业获得学士学位。现在国防科技大学网络与信息安全专业攻读硕士学位。研究领域为信息安全、数据分析。研究兴趣包括: 网络黑产治理、网络内容安全。Email: wangbo_wb@nudt.edu.cn

施凡 国防科技大学电子对抗学院副教授, 硕士生导师。青年科技英才, 国防科技大学学科领军人才, 国家重点研发计划首席青年科学家, 享受军队一类专业技术人才岗位津贴, 主要从事网络空间测绘、网络安全知识图谱构建、网络渗透测试等方面的研究, 主持和参与国家高新工程、国家重点研发以及军队级项目 10 余项, 获军队(省)级科技进步一等奖 2 项, 二等奖 4 项, 三等奖 6 项。