

基于组件分割的钓鱼 URL 检测方法

钟文康¹, 王 添², 张功萱³

¹网络空间安全学院 南京理工大学 南京 中国 210094

²信息工程学院 江苏财会职业学院 连云港 中国 222061

³计算机科学与工程学院 南京理工大学 南京 中国 210094

摘要 URL 作为钓鱼网站最直接也是最重要的特征, 利用深度学习的方法对分词后的 URL 字符序列进行特征提取, 可以极大的提升基于 URL 的钓鱼网站识别的准确率。将 URL 按照不同组件进行分割是 URL 常见的分词手段, 该方法能够对不同组件进行多粒度的特征判别, 但是这一方法未能在钓鱼网站的 URL 检测中得到有效应用, 尚缺乏深入的研究。此外, 现有的基于深度学习的钓鱼网站 URL 检测方法由于实验数据以及模型训练方法上的局限性, 在泛化能力和误报率方面仍存在不足, 难以满足真实环境中复杂的识别需求。为解决上述问题, 本文提出了一种基于组件分割的钓鱼 URL 检测方法: (1)该方法首先对 URL 的不同组件进行分割, 并对各组件依次进行字符级分词、截断填充及编码, 使得深度学习模型能够对不同组件采取不同层级的管理从而进行细粒度的特征判别。(2)为了避免卷积神经网络中采用的池化策略过于关注局部特征而忽视特征整体空间结构的问题, 本文所提方法将对融合后的各组件特征利用胶囊网络进一步提取。(3)在模型训练方法中引入对抗训练机制, 对多嵌入层进行独立对抗训练, 以满足模型对各组件的差异化处理, 从而进一步提升模型的泛化能力。最后, 在百万级的样本数据集中, 与现有的最先进的同类方法相比, 所提方法在钓鱼 URL 的识别准确率上提升 0.86%, 误报率降低 1.08%, F1-Score 提升 0.95%。

关键词 钓鱼 URL 检测; 胶囊网络; 对抗训练; 数据处理; 深度学习

中图分类号 TP391 DOI 号 10.19363/J.cnki.cn10-1380/tn.2025.01.10

Phishing URL Detection Method Based on Component Segmentation

ZHONG Wenkang¹, WANG Tian², ZHANG Gongxuan³

¹School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

²School of Information Engineering, Jiangsu College of Finance & Accounting, Lianyungang 222061, China

³School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Abstract As the most direct, important feature of phishing websites, feature extraction of URL character sequences after word segmentation can improve the accuracy of URL-based phishing detection using deep learning methods. Segmentation of URLs by components is a commonly used URL processing method that enables models to discriminate between the components at different granularities while this method has not been used in phishing URL detection, and the effectiveness of this processing still needs to be independently experimentally demonstrated. Due to the limitations of experimental data and model training methods, existing deep learning-based phishing URL detection methods still have shortcomings in terms of generalization ability and false alarm rate, which are challenging to meet the complex needs of real-world environments. To solve the above problems, this paper proposes a component-based segmentation method for phishing URL detection: (1) We first segment the URLs into different components, we then perform character-level word separation, truncation filling, and coding for each component so that the deep learning model can adopt different degrees of strict and fine-grained feature discrimination for different components. (2) To avoid the pooling strategy used in convolutional neural networks (CNNs), which focuses on local features and ignores the overall spatial structure of the features, the proposed method uses a capsule network (CapsNet) to extract the fused features of each component further. (3) The adversarial training mechanism is introduced in the model training method to conduct independent adversarial training for multiple embedding layers to satisfy the differentiation of the model for each component, further enhancing the generalisation capability of the model. Through extensive simulations, the result shows a 0.86% improvement in accuracy, 1.08% reduction in false alarm rate, and 0.95% improvement in F1-Score compared to existing state-of-the-art methods in a dataset of millions of samples.

通讯作者: 王添, 博士, 讲师, Email: wangtian@jscfa.edu.cn.

本课题得到国家自然科学基金(No. 62272232), 江苏省自然科学基金青年基金项目(No. SBK2024041254), 江苏省高等学校自然科学研究面上项目(No. 24KJB520002), 连云港市科技计划项目(No. JCYJ2328), 江苏财会职业学院科研启动基金 (No. 2023GC06)资助

收稿日期: 2023-02-27; 修改日期: 2023-04-25; 定稿日期: 2024-11-20

Key words phishing URL detection; capsule networks; adversarial training; data processing; deep learning

1 引言

近年来,随着互联网技术的快速发展和普及,互联网的使用量呈指数级增长,与此同时,人们也越来越容易遭受来自互联网的攻击。网络钓鱼便是最危险的网络攻击之一,它利用社会工程和技术手段窃取用户的个人身份信息和金融账户以实施犯罪。根据反网络钓鱼工作组(The anti-phishing working group, APWG)在 2022 年第三季度的网络钓鱼活动趋势报告^[1]显示,在该季度 APWG 共记录 1270883 起网络钓鱼攻击,刷新了由上一季度创下的 1097811 起网络钓鱼攻击记录。由钓鱼网站引发的网络钓鱼攻击事件逐年增长,可见钓鱼网站出现之频繁,增长之快,因此,需要先进的算法来及时、有效的检测钓鱼网站。

通常情况下,网络钓鱼攻击起始于受害者点击了攻击者精心制作的钓鱼网站 URL,即使用户打开钓鱼 URL 后识破该网站为钓鱼网站并将其关闭,钓鱼网站也很可能会携带病毒或恶意代码影响受害者主机。在用户点击钓鱼 URL 之前对其进行检测并提示就显得尤为重要了。根据用于检测的数据类型不同,钓鱼网站检测方法可以分为四类:基于 URL 文本^[2]、基于网站主机信息^[3]、基于网站视觉特征^[4]和基于网站内容^[5],基于 URL 文本的方法明显相较于基于网站主机信息、视觉特征及网站内容的检测方法更快速,因为这种方法无需进行页面解析和网络查询,事实上由于钓鱼网站存活时间极短^[6]且存在多种躲避检测行为,对钓鱼网站的内容及主机信息进行大规模收集也是十分耗时费力的工作。

近年来虽然越来越多研究人员提出对仅使用钓鱼 URL 作为特征来源的检测方法的质疑,但是由于钓鱼网站的制作成本较低、收益不高且人们反钓鱼意识逐渐增强等情况,钓鱼攻击者往往会批量生成大量粗制的钓鱼网站,其 URL 与正常网站的 URL 存

在较大的字符搭配差异,例如 URL 长度、特殊符号的使用、数字的使用等,尤其是越为知名的企业网站与其仿冒的钓鱼 URL 差异越大。神经网络通过对大量钓鱼 URL 和正常 URL 的字符特征提取,与自然语言处理领域中文本分类任务类似,模型能够学习到正常 URL 与钓鱼 URL 之间字符搭配的差异等特征,大量实验结果表明了这种方法的有效性。钓鱼 URL 检测作为钓鱼网站检测的一大分支,钓鱼 URL 检测技术的研究能够促进钓鱼网站甚至其他恶意网站检测的发展。

目前许多基于深度学习的钓鱼 URL 检测方法^[7-10]主要是利用对 URL 字符序列特征的自动提取,具体而言是通过对整个 URL 进行字符级分词,获得每个字符的嵌入向量后对得到的整个 URL 的嵌入矩阵进行特征提取,虽然取得了较好的表现,但仍存在一些问题:①在数据方面,大部分研究所采用的实验数据集存在数据量较少、数据源少、数据 URL 格式单一的情况,Abdillah 等人^[11]调查显示,在 68 篇较高水平的钓鱼攻击检测论文中仅有 13%的研究采用了百万级以上的实验数据,深度学习模型需要大量数据的支持,较少的实验数据极易导致模型过拟合,而且 URL 的字符特征极为丰富,某些数据集在收集过程中可能会偏重于 URL 的某些特征,在数据量少的数据集上进行实验很可能会取得过于乐观的结果。②许多研究所提方法虽然能够在自身的数据集中取得较好的检测效果,但是在更换数据集后,所取得检测效果降低较为明显,难以在实际环境中取得较好的效果,也即模型的泛化能力不足,并不能十分良好的学习到钓鱼 URL 本质的检测特征。③现有检测方法中的 URL 处理方式主要是对整个 URL 进行字符级^[12]或单词级^[13]的分词编码,得到整个 URL 的嵌入矩阵表示,却忽视了 URL 的结构问题,一个典型的 URL 由 7 个组件构成,其结构如图 1 所示。

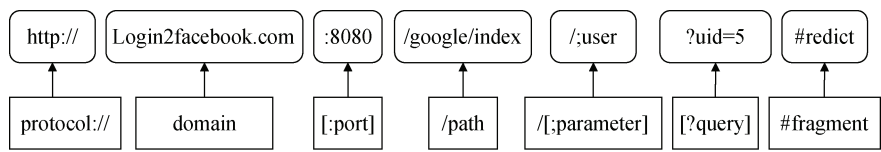


图 1 URL 的典型结构图
Figure 1 Typical structure of a URL

其中,域名(Domain)组件由于其唯一性和代表性,正常网站的域名是更为规范,如极少使用数字、

特殊字符,域名长度适中,字符搭配合理等,是与钓鱼 URL 区别最大的部分。而其他如路径(Path)等组件,

则完全由网站开发者确定, 其受规范的制约较弱, 但同样存在一些能够区分正常 URL 与钓鱼 URL 的特征。对整个 URL 进行字符级或单词级的分词很可能导致模型无法区分人为构造的域名、路径、参数等组件, 而对实际上的各组件采用相同的粗粒度的特征判别, 考虑如图 2 所示的两个示例链接。

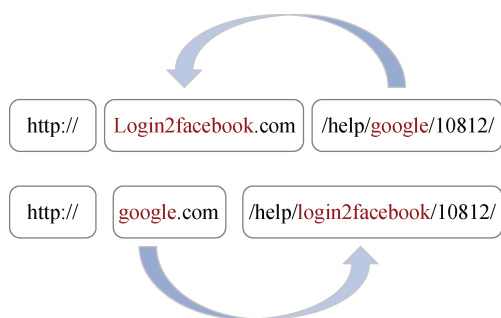


图 2 更换示例 URL 中域名与路径的部分内容
Figure 2 Replace part of the domain name and path in the example URL

很显然在交换了域名中的 login2facebook 到路径中之后, 网站已经发生了变化, 由不可信网站变成了可信网站, 但是在现有模型的视角下, 可疑特征 login2facebook 仍然存在, 那么将很可能导致误判。这样的结果在现实中是无法接受的。

针对以上可能存在的问题, 本文提出一种基于组件分割的钓鱼 URL 检测方法(Phishing URL detection method based on component segmentation, CS-PUD), 在与多项同类深度学习的方法进行同数据集实验后, 实验结果表明了本文所提方法的有效性。

具体而言, 本文的主要贡献如下。

1) 提出使用能够保留特征整体空间结构的胶囊网络代替卷积神经网络对 URL 字符序列进行特征自动提取的方法, 并在钓鱼 URL 检测领域引入了对抗训练机制提升了钓鱼 URL 检测模型的鲁棒性和泛用性。

2) 对 URL 进行组件分割的必要性进行了实验论证, 同时结合多组对比实验, 探索了 URL 各组件对钓鱼 URL 检测的贡献情况。

3) 从多个公开数据源收集并构建了 180 多万条正常 URL 和钓鱼 URL 比例近似 1:1 的多源数据集, 还使用了三个公开钓鱼网站检测研究数据集进行对比实验, 以保证实验过程中公平、真实的钓鱼 URL 检测环境, 并将所提方法与现有的最新方法进行对比, 实验结果表明本文所提方法优于对比方案。

2 相关工作

目前国内外的研究人员已经提出了多种针对钓

鱼网站的检测方法与检测技术, 流行的钓鱼网站检测方法主要包括黑/白名单、基于传统机器学习和基于深度学习的检测方法。

基于黑/白名单的钓鱼网站检测法以其极低的误报率, 便于应用的优点受到较为广泛的使用^[14], 但是其致命的弱点是它无法及时识别新出现的钓鱼网站 URL^[15]。Jain 等人^[16]开发了一个基于白名单的恶意 URL 检测系统, 它可以阻止对所有不在该名单上的网站的访问。与此相对的是黑名单系统^[14]则更为常见, 因为在同样低误报率的情况下, 黑名单系统能够允许对更丰富的网站的访问。

为了克服黑/白名单检测法的缺陷, 研究者们一直在寻求使用机器学习方法来检测未报告的钓鱼网站。Sahingoz 等人^[17]利用随机森林(Random forest, RF)算法在 Ebbu2017^[18]数据集中获得了 97.98% 的检测准确率。Daeeef 等人^[19]使用了多种机器学习算法, 根据 URL 的 n -gram 特征对 URL 进行分类, 测试结果检测准确率为 93%。YU Enze 等人^[3]根据网站所采用协议、文本关键字置信度、钓鱼类词汇相似度、URL 文本特征等特征, 最终经随机森林分类模型进行分类, 在包含 10975 个网站的数据集中平均准确率为 99.6%。Fu. A. Y 等人^[20]使用网页源文件将网络钓鱼网页的视觉外观与其潜在目标的视觉外观进行比较来判断是否为钓鱼网站, 使用的数据集包含 10281 个可疑网站, 达到 99.87% 的准确率和 88.88% 的召回率。

深度学习以其自动提取数据中的潜在特征的优点受到学术界与工业界的广泛应用, Yuan J 等人^[7]提出了一种分别利用胶囊网络(Capsule network, CapsNet)和独立循环神经网络(Independent recurrent neural network, IndRNN)的并行联合神经网络提取 URL 中的视觉和语义特征的检测方法, 在 66017 个 URL 的数据集上取得了 99.78% 的准确率。Aljofey 等人^[12]将 URL 作为原始输入, 利用多层卷积神经网络(Convolutional Neural Networks, CNN)对 URL 的字符序列特征进行学习训练, 最终分别在 3 个数据集中达到 98.58%、95.46% 和 95.22% 的准确率, 表明 CNN 对 URL 字符序列有良好的学习能力。Al-Alyan 等人^[8]同样使用基于卷积神经网络模型仅依赖 URL 的检测方法。最终在 2333852 条 URL 的数据集中取得了 95.78% 的准确率。Peng 等人^[9]首先利用 CNN-LSTM 的网络结构快速对 URL 进行分类, 通过 softmax 得出钓鱼网站概率。之后将 URL 统计特征、网页代码特征、网页文本特征和钓鱼网站概率合并为多维特征, 最后利用 XGBoost 进行分类, 在 2010779 个 URL 的数据集中取得了 98.99% 的准确

率。Bu 等人^[10]利用卷积自动编码器对 URL 进行字符级特征提取, 在 222541 个 URL 的数据集中取得了 96.1% 的预测准确率。

虽然现有的基于深度学习的分类模型均取得了较好的检测效果, 但除了更换新的特征提取模型外, 数据处理方式以及模型训练方式仍然存在提升空间, 本文提出的 CS-PUD 方法, 通过人为的对 URL 按照组件进行分割筛选以及胶囊网络的引入能够提升模型对 URL 的字符特征提取能力, 同时使用对抗训练方法替代传统模型训练方法, 提升了模型的鲁棒性和泛化能力。

3 基于组件分割的钓鱼 URL 检测方法

在本节中, 本文将介绍对 URL 进行组件分割的方法、字符级分词方式、模型结构以及对抗训练过程, 本文所提出的 CS-PUD 方法总流程如图 3 所示。

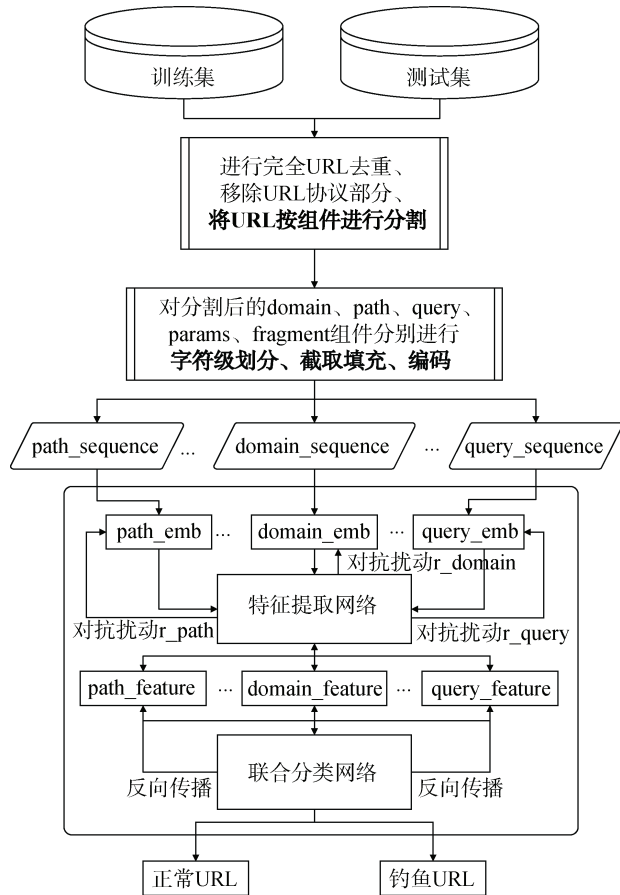


图 3 CS-PUD 方法流程
Figure 3 CS-PUD method flow

3.1 URL 组件分割

一个典型的 URL 包含 7 个组件, 各个组件重要性不同、功能不同, 将 URL 分割为不同的组件, 进行组件级的数据处理, 能够使得模型对各组件采取

不同的特征判别, 进行差异化处理, 示例 URL 的分割结果如表 1 所示。

表 1 示例 URL 的组件分割结果
Table 1 Component segmentation results for example URLs

原始 URL	Login2facebook.com/Fd/cmd
URL 组件分割	['Login2facebook.com', 'Fd/cmd', ',', '/', '']

由于钓鱼 URL 制作成本低廉, 其形式多样且存在躲避解析行为, 对 URL 进行组件分割需要考虑到各种可能存在的情况, 组件分割算法如算法 1 所示。

算法 1. URL 组件分割算法.

输入: 原始 URL 数据 u

输出: 对 u 的组件分割结果 res

- 1) 初始化 $count=0$
- 2) WHILE u 能够被正常解析 且 $count<5$ DO
- 3) IF u 包含协议部分 THEN
- 4) 解析并清除 u 的最外层协议
- 5) ELSE
- 6) $u="http://"+u$
- 7) $count+=1$
- 8) END WHILE
- 9) IF $count==5$ THEN
- 10) $res = []$ /* u 无法被正常解析分割 */
- 11) ELSE
- 12) $res = [u.domain, u.netloc, u.path, u.params, u.query, u.fragment]$ /* u 的各组件 */
- 13) END IF
- 14) RETURN res

3.2 字符级分词

由于 URL 并未严格要求采用单词进行构建, URL 中存在大量的非单词字符组合, 对 URL 采用单词级分词方式将产生庞大的语料库, 且难以处理未出现过的词汇。字符级分词方式将 URL 划分为单个字符, 能够避免产生巨量的语料库且能够充分体现钓鱼 URL 中的字符级变化, 便于模型捕捉到形状相似字符的替换。为使用户无法直接分辨钓鱼 URL, 攻击者常常会使用形状近似的字符进行替换, 如使用数字 1 替换字母 L 的小写形式 l 等, 字符级分词方式也是较多研究所采用的方式。表 2 中展示了按照字符级分词的结果。表 3 列出了本文采用的字符级分

表 2 示例 URL 的字符级分词方式

原始 URL	Login2facebook.com/Fd/cmd
字符级分词	L o g i n 2 f a c e b o o k . c o m / F d / c m d

表 3 字符编码映射表

Table 3 Character code mapping table

字符	编码
abcdefghijklmnopqrstuvwxyz	1-26
ABCDEFGHIJKLMNOPQRSTUVWXYZ	27-52
0123456789	53-62
.,:!\?:"'"/_\@#\$\$%^&*~`'+-=<>(){}	63-94
<UNK>	95
<PAD>	0

表 4 示例 URL 的完整处理过程

Table 4 The full processing of the example URL

原始 URL	Login2facebook.com/Fd/cmd
组件分割	['Login2facebook.com', '/Fd/cmd', ',', ',', ',']
字符级分词	['L','o','g','i','n','2','f','a','c','e','b','o','o','k','.','c','o','m','.','/','F','d','/','c','m','d','.','.',',',',',',']
编码	[[[38], [15], [7], [9], [14], [55], [6], [1], [3], [5], [2], [15], [15], [11], [65], [3], [15], [13]], [[72], [32], [4], [72], [3], [13], [4]], [], [], []]

词方式所使用的字符编码映射表, 其中<UNK>标记表示未出现在映射表中的字符, <PAD>则表示对不

足指定长度的序列进行填充时的填充内容。表 4 展示了示例 URL 经过组件分割及字符级分词后的编码结果。

在根据表 3 中的字符编码映射表获得 URL 的编码序列后, 由于神经网络只能接受定长的序列, 还需要对每条 URL 所得的各组件序列进行截断填充, 即对于各组件序列, 若该序列超过 L 个单位长度则对超出的尾部内容进行删除, 若该序列小于 L 个单位长度则将序列填充 0 直至序列长度达到 L 个单位长度。

3.3 模型结构设计

由于模型包含多部分输入, 为分别从组件角度以及 URL 整体角度提取钓鱼 URL 特征, 本文将所设计的模型网络分为组件特征提取网络以及联合分类网络, 模型结构如图 4 所示。组件特征提取网络采用 2 层 CNN1D 及 1 层 Bi-LSTM 构成, 这种简单的特征提取网络在多项研究中证明了其特征提取的有效性。在对各组件进行特征提取后, 联合分类网络会将各组件的深层特征进行组合并进一步特征提取。

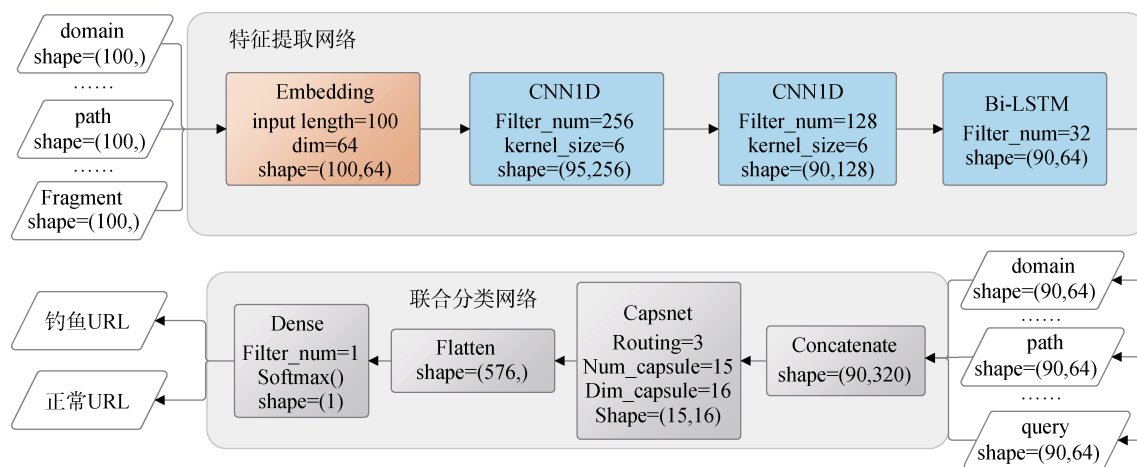


图 4 CS-PUD 方法使用的模型详细结构

Figure 4 Detailed structure of the model used by the CS-PUD method

胶囊网络最开始在计算机图像领域取得了较好的应用, Zhao 等人^[21]在首次将胶囊网络应用于文本分类时, 便取得了优于卷积神经网络和循环神经网络的分类效果。Yuan J 等人^[7]首次将胶囊网络应用于钓鱼 URL 视觉特征的提取。本文结合胶囊网络在文本分类领域以及钓鱼 URL 检测领域的应用, 使用胶囊网络对联合后的深层字符序列特征进行进一步的提取。

胶囊网络使用向量胶囊来替代卷积神经网络中的神经元, 使用动态路由来替代池化操作, 使用 Squash 函数来替代 Relu 激活函数。这种将神经元的

标量输出转为向量输出提高了表征能力, URL 的各组件之间存在固定的顺序, 胶囊网络的向量输出能够很好表示 URL 的某个特征, 同时还能表示这个特征的位置等物理特征。

3.4 多 Embedding 层同时独立对抗训练

模型的泛化能力对于恶意检测领域而言十分重要, 是其是否能够良好应用的标准。由于数据收集的局限性, 对模型进行训练的数据难以做到与实际环境中的数据同分布。良好的检测模型应当具备在有限的数据集中尽可能充分地学习到检测问题本质的特征, 并正确预测未在数据集中出现过的样本。基于

对抗训练的思想, 经过对抗训练后, 检测模型能够具备检测钓鱼 URL 相似变体的能力, 如图 5 所示, 对于钓鱼网站 login2faceb00k.com, 经过加入扰动后使得模型能够检测出变体 login2facebook.com, 这是泛化能力的体现。

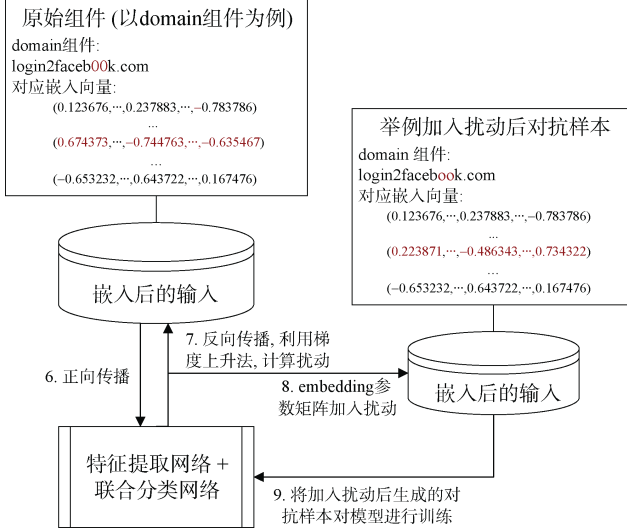


图 5 钓鱼 URL 检测中的对抗训练

Figure 5 Adversarial training in phishing URL detection

与自然语言处理领域中对抗训练的思路一致,

在 Embedding 层中添加扰动, 得到最鲁棒的 Embedding 向量。CS-PUD 方法的模型中包含多部分输入, 各个组件的输入对应专属的 Embedding 层。对各个组件的 Embedding 层进行独立的对抗训练, 也即对各 Embedding 层分别计算并添加扰动, 各组件在输入上和钓鱼特征上存在差异, 独立对抗训练能够满足模型对各组件的差异化处理, 大幅度增强模型对钓鱼网站 URL 整体的抗干扰能力, 如图 6 所示。

对抗训练能否取得最好的效果取决于能否得到最佳的对抗扰动 Δx , 基于多 Embedding 层独立对抗训练的思想, 本文对传统的对抗训练方法 FGM^[22]、PGD^[23]、FreeAT^[24]、FreeLB^[25]均进行了改进尝试, 最终在 FreeLB(Free Large-Batch)方法中取得了最佳的检测效果, 改进后的 FreeLB 对抗训练算法如算法 2 所示。

为保证能够针对各组件 Embedding 层进行独立的对抗训练同时尽量减少每批次数据的训练次数, 在每批次训练的 K 步迭代上升中加入了对各组件 Embedding 层的独立处理, 分别利用模型的梯度对各 Embedding 层计算扰动, 并将其保存在临时数组 δ_{temp} 中, 最后加入到各组件原始输入中构成对抗样本, 使得仅针对单 Embedding 层的 FreeLB 算法具备对多 Embedding 层模型进行对抗训练的能力。

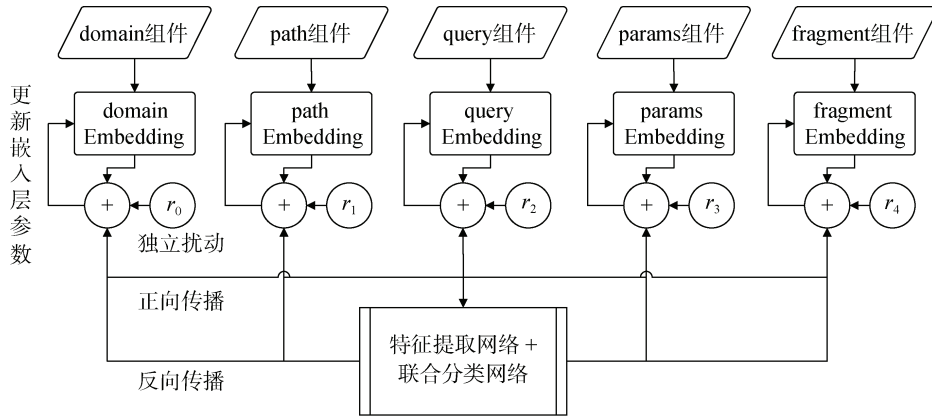


图 6 对各 Embedding 层进行独立对抗训练

Figure 6 Independent adversarial training for each Embedding layer

算法 2. 多 Embedding 层独立对抗训练.

输入: 原始训练样本 $X=\{(Z, y)\}$, 扰动界限 ε , 学习率 τ , 上升迭代次数 K , 上升步幅 α

输出: NULL

1) 初始化 θ

2) FOR $epoch = 1 \dots N_{ep}$ DO

3) FOR minibatch $B \subset X$ DO

4) $g_0 \leftarrow 0$

$$5) \quad \delta_0 \leftarrow \frac{1}{\sqrt{N_\delta}} U(-\varepsilon, \varepsilon)$$

6) FOR $t = 1 \dots K$ DO

7) 计算当前模型参数 θ 的梯度

8)

$$g_t \leftarrow g_{t-1} + \frac{1}{K} \mathbb{E}_{(Z, y) \in B} [\nabla_{\theta} L(f_{\theta}(X + \delta_{t-1}), y)]$$

$$9) \quad \delta_{temp} = []$$

```
10)      FOR  $j$  in [domain, path, query, params,
fragment] DO
11)          通过梯度上升更新扰动  $\delta_{t,j}$ 
12)           $g_{adv,j} \leftarrow \nabla_{\delta_j} L(f_{\theta}(X + \delta_{t-1,j}), y)$ 
13)           $\delta_{t,j} \leftarrow \Pi_{\|\delta\|_F \leq \epsilon} (\delta_{t-1,j} + \alpha \cdot g_{adv,j} / \|g_{adv,j}\|_F)$ 
14)           $\delta_{temp} = \delta_{temp} + \delta_{t,j}$ 
15)      END FOR
16)       $\delta_t = \delta_{temp}$ 
17)  END FOR
18)       $\theta \leftarrow \theta - \tau g_K$ 
19)  END FOR
20) END FOR
```

4 实验评估与分析

4.1 实验环境与数据集

实验环境和配置信息如下:

硬件配置: Windows 10 21H2, 16GB 内存, AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz, GPU NVIDIA GeForce RTX 3060 Laptop GDDR6 6GB,

256GB SSD + 512GB SSD 硬盘;
软件环境: python 3.8.12、tensorflow-GPU 2.4.1;
数据集有效性验证. 为了检验 CS-PUD 方法的有效性, 尽可能模拟在现实中的检测环境, 本文从互联网中广泛收集可信数据, 既从公开钓鱼 URL 研究数据集中收集, 同时也从知名钓鱼 URL 发布网站中爬取最新数据, 最终将数据集分为 4 个数据集, 具体数据情况如表 5、表 6 所示。其中数据集 1 为多源数据集, 包含三个公开知名钓鱼网站相关数据集、从知名反钓鱼网站社区 Phishtank 爬取的最新钓鱼 URL 以及 Google 常见单词搜索结果前 500 条记录的正常 URL。相较于另外几个公开钓鱼网站检测研究数据集 ISCX-URL^[27]、PSU^[28]和 PhishStorm^[29], 数据集 1 拥有时间上更新以及数据量更大的优势, 总共包含 939512 条正常 URL 和 794692 条钓鱼 URL, 更具代表性。

同时为保证实验结果能够体现模型对未知钓鱼 URL 的检测结果, 各数据集在进行实验时都将在训练前按照 7 : 1 : 2 的比例设为训练集、验证集和测试集, 所有的评价指标均在测试集中取得。

表 5 数据集构成
Table 5 Dataset composition

数据集名称	收集年份	数据源	合法 URL	钓鱼 URL
数据集 1	2019	Kaggle:Malicious And Benign URLs ^[26]	345738	0
	2022	Google Search: The first 500 URLs of each search result	165671	0
	2022	PhishTank: Manual collection	0	100019
	2021	Kaggle: Malicious URLs dataset	428103	94111
	2022	Github: Phishing.Database.ALL-phishing-links	0	600562
数据集 2	2016	ISCX-URL2016 ^[27]	35378	9965
数据集 3	2019	Kaggle:Phishing Site URLs ^[28]	392924	156422
数据集 4	2014	PhishStorm: Detecting Phishing with Streaming Analytics ^[29]	48009	47902

表 6 各数据集中 URL 各组件长度统计
Table 6 Length statistics for each component of the URL in each dataset

数据集	domain		path		params		query		fragment	
	平均长度	75%数据长度	平均长度	75%数据长度	平均长度	75%数据长度	平均长度	75%数据长度	平均长度	75%数据长度
数据集 1	20.60	24.00	28.26	39.00	0.06	0.00	9.54	0.00	0.04	0.00
数据集 2	14.68	15.00	66.98	85.00	0.01	0.00	19.29	6.00	0.01	0.00
数据集 3	17.15	19.00	27.11	37.00	0.01	0.00	6.97	0.00	0.03	0.00
数据集 4	22.31	22.00	25.83	33.00	0.06	0.00	15.90	0.00	0.08	0.00

4.2 评价指标与超参数设置

本文设定钓鱼网站为阳性(Positive)样本, 合法网站为阴性(Negative)样本, TP 、 FP 、 TN 和 FN 分别表示真阳性、假阳性、真阴性和假阴性的数量。由

于钓鱼 URL 检测是一种二分类实验且合法网站被分类为钓鱼网站往往是现实中更加难以接受的结果, 本文使用钓鱼网站检测领域广泛使用的检测准确率(Accuracy)、真阳性率(TPR , True positive rate)、假阳

性率(*FPR*, False positive rate)以及 *F1-Score* 作为评价指标。真阳性率也即检测率定义为检测到的钓鱼网站数占钓鱼网站总数的比例又称召回率, 由于漏报率与召回率呈负相关, 召回率越高漏报率便相应越低, 本文便不再对漏报率进行额外说明。假阳性率即误报率为错误分类为钓鱼网站的合法网站数占有合法网站数的比例, *F1-Score* 同时兼顾了模型的精确率和召回率, 能够对模型进行综合性评价。一个理想的钓鱼网站检测方法必须同时具有高检测准确率、高召回率、高 *F1-Score* 和低误报率。这四个标准的计算公式如下:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$TPR(recall) = \frac{TP}{TP + FN} \quad (2)$$

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

$$F1-Score = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

在 CS-PUD 方法中主要的超参数包含 URL 各组件的截取长度以及词嵌入层的嵌入维度, 首先对于 URL 组件分割后各组件的截取长度, 从表 6 中可观察到各组件的平均长度均在 40 以内。为尽量减少因截取而导致的信息损失, 本文将各组件截取的长度分别设置为 60、80、100, 均远大于数据集中各组件 75% 数据的长度, 同时为避免大量的填充 0 可能对模型造成的干扰, 本文对填充 0 进行了掩码处理, 模型将对填充的 0 进行计算屏蔽。除此之外嵌入矩阵的嵌入维度同样会影响模型的预测能力, 本文将嵌入层的 *embedding_dim* 参数分别设置 64、128 的取值参与测试。实验结果如表 7 所示, 当嵌入维度为 64, 截取长度为 100 时准确率、*F1-Score* 最高, 误报率较低, 检测效果最好, 后续实验均将采用这个组合, 模型训练的其他参数如表 8 所示。

表 7 在数据集 1 上不同截取长度、嵌入维度的实验结果

嵌入维度	截取长度	Accuracy/%	TPR/%	FPR/%	F1-Score/%
64	60	98.42	98.04	1.25	98.18
64	80	98.40	97.87	1.16	98.15
64	100	98.43	97.87	1.07	98.18
128	60	98.39	97.83	1.14	98.13
128	80	98.39	97.66	1.00	98.13
128	100	98.39	97.84	1.15	98.13

表 8 模型训练参数

参数名	数值	说明
batch_size	64	每批次样本数量
epoch	30	训练批次
learning_rate	0.001	学习率
routing	3	动态路由迭代次数
num_capsule	15	胶囊数量
dim_capsule	16	胶囊向量维度
epsilon	1	对抗训练扰动范围
alpha	0.2	梯度上升步幅

4.3 实验结果与分析

4.3.1 对比实验

首先对 CS-PUD 方法的有效性进行验证, 由于钓鱼网站存活时间极短, 且存在多种躲避解析行为, 因此难以对使用了包含除 URL 外其他特征的深度学习方法进行对比实验。最终本文选择将 CS-PUD 方法与 Al-Alyan^[8]、Ali Aljofey^[12]、Ren^[30]、Huang^[31]、Yuan J^[7]等人的方法进行对比, 这几项研究均是仅从 URL 中提取相关特征的最新的深度学习检测方法, 同时由于 Al-Alyan^[8]、Yuan J^[7]等对比方法已与如随机森林算法等传统机器学习方法进行了对比实验, 且均取得了比传统机器学习方法更好的效果, 此处便不再与传统机器学习方法进行对比。

在不同数据集上的对比实验结果如表 9、表 10、表 11、表 12 所示, 能够发现相较同类仅基于钓鱼

表 9 数据集 1 上的对比实验结果

Table 9 Results of comparison experiments on dataset 1

Methods	Accuracy/%	TPR/%	FPR/%	F1-Score/%
Al-Alyan ^[8]	96.43	95.76	3.02	96.08
Aljofey ^[12]	96.93	96.18	2.44	96.62
Ren ^[30]	97.04	96.63	2.75	96.78
Huang ^[31]	97.10	96.69	2.41	96.73
Yuan J ^[7]	97.57	96.90	2.15	97.23
CS-PUD	98.43	97.87	1.07	98.18

表 10 数据集 2 上的对比实验结果

Table 10 Results of comparison experiments on dataset 2

Methods	Accuracy/%	TPR/%	FPR/%	F1-Score/%
Al-Alyan ^[8]	99.45	98.79	0.28	99.07
Aljofey ^[12]	99.46	98.66	0.19	98.99
Ren ^[30]	99.54	99.02	0.35	99.24
Huang ^[31]	99.47	99.14	0.37	99.16
Yuan J ^[7]	99.69	99.36	0.15	99.38
CS-PUD	99.87	99.73	0.06	99.77

表 11 数据集 3 上的对比实验结果
Table 11 Results of comparison experiments on dataset 3

Methods	Accuracy/%	TPR/%	FPR/%	F1-Score/%
Al-Alyan ^[8]	97.56	93.74	1.33	94.54
Aljofey ^[12]	98.20	94.77	0.80	95.96
Ren ^[30]	98.12	93.93	0.96	95.76
Huang ^[31]	98.27	94.64	0.71	95.88
Yuan J ^[7]	98.39	94.76	0.55	95.91
CS-PUD	98.86	95.14	0.38	96.63

表 12 数据集 4 上的对比实验结果
Table 12 Results of comparison experiments on dataset 4

Methods	Accuracy/%	TPR/%	FPR/%	F1-Score/%
Al-Alyan ^[8]	97.89	97.72	1.93	97.89
Aljofey ^[12]	97.79	97.33	1.75	97.78
Ren ^[30]	97.91	96.78	0.96	97.89
Huang ^[31]	98.26	97.31	1.07	97.95
Yuan J ^[7]	98.41	97.62	0.93	98.01
CS-PUD	98.93	98.58	0.86	98.70

表 13 数据集 1 上不同组件组合的实验结果
Table 13 Experimental results for different combinations of components on dataset 1

实验编号	组件构成	Accuracy/%	TPR/%	FPR/%	F1-Score/%
1	domain-path-param-query-fragment	98.43	97.87	1.07	98.18
2	domain-path-query-fragment	98.41	98.00	1.38	98.16
3	domain-path-param-query	98.40	98.09	1.28	98.16
4	domain-path-query	98.40	98.27	1.58	98.14
5	domain-path-param	98.17	97.90	1.56	97.87
6	domain-path-fragment	98.14	98.16	1.88	97.84
7	domain-path	98.13	97.86	1.60	97.82
8	domain-path-param-fragment	98.07	97.93	1.78	97.77
9	domain-query	93.44	92.61	5.73	93.38
10	domain-param-query	93.35	93.33	6.62	93.34
11	domain-query-fragment	93.29	92.18	5.60	93.20
12	domain-param-query-fragment	93.18	92.90	6.53	93.16
13	domain-param-fragment	92.78	92.79	7.23	92.77
14	domain	92.85	92.70	6.99	92.84

对比两种不同处理方式, 能够发现进行组件分割相比不进行组件分割的实验, 准确率提升 0.39%, 误报率降低 0.51%, F1-Score 提升 0.3%。说明对 URL 进行组件分割能够在一定程度上提升模型对钓鱼 URL 的检测效果。

(2) URL 组件选取的影响

对 URL 进行组件分割后, 将主要产生 5 个不同

URL 的检测方法, 本文提出的 CS-PUD 方法在准确率、召回率、误报率以及 F1-Score 上均取得了最好的检测效果, 即便在百万级的数据集 1 中, 预测准确率达 98.43%、F1-Score 达 98.18%, 均高于其他方法; 而误报率也是大幅度低于其他方法, 这表明本文所提出的 CS-PUD 方法能够有效的对钓鱼网站 URL 进行检测。

4.3.2 特征消融实验

为进一步探究 CS-PUD 方法中 URL 组件分割、URL 组件的选取、胶囊网络、改进的对抗训练等提升方法对钓鱼 URL 检测的提升效果, 设计了多组特征消融实验。

(1) URL 组件分割的影响

进行 URL 组件分割是 CS-PUD 方法区别于其他同类方法的关键步骤之一, 本文认为模型需要对 URL 的各个组件进行不同严格程度和粒度的特征判别。为验证 URL 组件分割的有效性, 本文分别对进行 URL 组件分割和不进行 URL 组件分割的方法进行对比实验, 其中不进行 URL 组件分割的实验采用截取长度为 200, 嵌入维度为 128 的组合, 其他参数及结构与进行 URL 分割实验一致。实验结果如图 7 所示。

功能的组件, 若将所有组件均作为输入, 虽然能够最大程度上的保留 URL 信息, 但是其模型参数及训练时间会大幅增加。为探索 URL 不同组件对钓鱼 URL 检测的贡献, 分别设计了多个不同组件的组合实验。

实验结果如表 13 所示, 所有组件均参与检测能够最大程度的利用信息, 也能够取得最佳的检测效

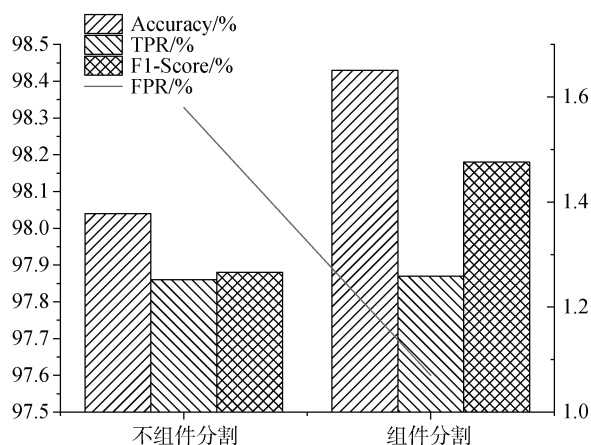


图7 数据集1上进行不同URL组件分割的对比实验
Figure 7 Comparative experiments with different URL component segmentation on dataset 1

果。还能观察到 1-4 号实验与其他剩余实验对比, domain-path-query 组合能够最大程度的保证模型准确率的同时减少模型的输入,这也与表 6 中这三部分组件的内容长度占比较高相印证。从第 7 组和第 9 组、第 8 组和第 12 组实验的结果对比来看, path 组件对检测的贡献高于 query 组件。从第 4 组与第 5 组、第 2 组与第 8 组实验的结果对比来看, query 组件的检测贡献高于 parameter 组件。而从第 2 组与第 3 组、第 5 组与第 6 组、第 10 组与第 11 组实验的结果对比来看, parameter 组件与 fragment 组件的检测贡献相差不大。最终本文能够得出各组件对钓鱼 URL 检测贡献程度: domain>path>query>parameter \geq fragment。

(3) 胶囊网络的提升效果

为探究胶囊网络对各组件深层特征的提取和对物理位置的表征效果,此处数据集 1 上设置了多组对比实验,分别使用全连接神经网络、一维卷积神经网络对胶囊网络进行替换,其他仍保持不变。全连接神经网络、一维卷积神经网络及胶囊网络的主要参数如表 14 所示。

表 14 数据集 1 上的对比实验结果

Table 14 Main parameters of the relevant layers

所使用网络	主要参数
全连接神经网络	units=16, activation=relu
一维卷积神经网络	Filters=16, activation=relu
胶囊网络	Num_capsule=15, Routing=3, Dim_capsule=16

实验结果如图 8 所示,对比全连接神经网络和一维卷积神经网络,胶囊网络以其使用向量表征深层特征、动态路由等特点,模型的泛化能力得到了提升,在 3 种神经网络中得到了最佳的检测效果,对比

一维卷积神经网络,准确率提升 0.16%,误报率降低 0.36%,F1-Score 提升 0.16%。

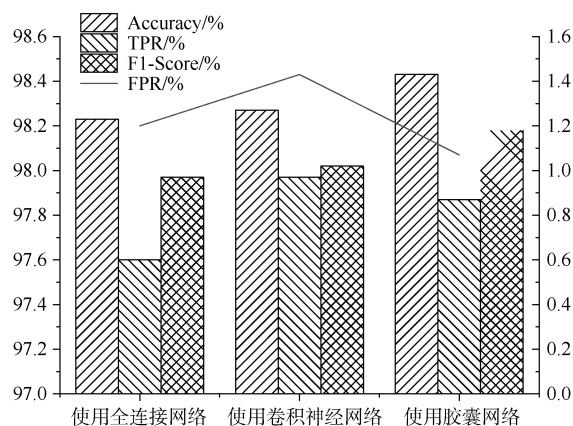


图8 数据集1上不同联合分类网络的对比实验
Figure 8 Comparison experiments of different joint classification networks on dataset 1

(4) 对抗训练的提升效果

为检验对抗训练是否能够提升钓鱼 URL 检测的效果,以及不同对抗训练方法对检测模型的影响。此处设计了多组对比实验,分别对不进行对抗训练、使用 FGM、PGD、FreeAT、FreeLB 改进方法进行了对比实验。

实验结果如图 9 所示,进行对抗训练提升了模型对钓鱼 URL 的检测准确率,降低了误报率,而 FGM 与 PGD 的提升效果接近,但均低于 FreeAT 和 FreeLB 方法,最终本文在 FreeLB 训练方法中取得了最佳的效果,可见对抗训练能够在一定程度上提升模型的泛化能力。

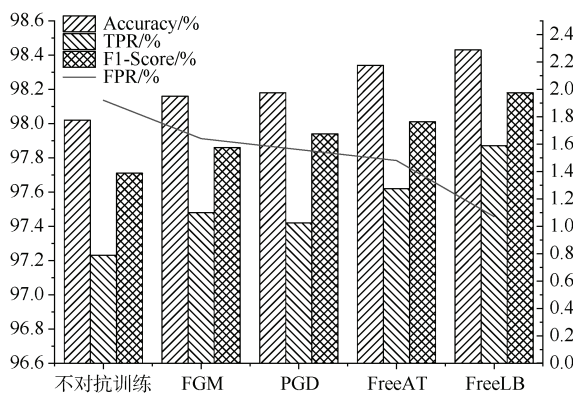


图9 数据集1上不同对抗训练方式的对比实验
Figure 9 Comparison experiments of different adversarial training methods on dataset 1

4.3.3 模型泛化能力验证实验

(1) 对未知钓鱼网站的预测能力

模型的泛化能力是本文主要关注以及提升的目的

标, 泛化能力是模型对未知数据的检测能力, 检测准确率越高说明模型的泛化能力越强。为进一步分析模型的泛化能力, 本文提取了 Openphish^①最新捕获的钓鱼网站数据, 包含 500 条全新的钓鱼网站 URL。Openphish 是一个全自动的独立网络钓鱼情报平台, 其报告的数据均为钓鱼网站的 URL, 被广泛用于钓鱼网站检测研究^[32]。此处利用表 9 中的方法分别对模型未知的 Openphish 数据进行预测, 预测结果如图 10 所示, 面对新的未出现过的数据, 所有方法均表现良好, 能够预测正确大多数的样例, 其中 CS-PUD 表现最佳, 仅有 2 条样例无法预测正确, 次优的 Yuan J 等人^[7]的方法则有 8 条 URL 无法进行正确预测。

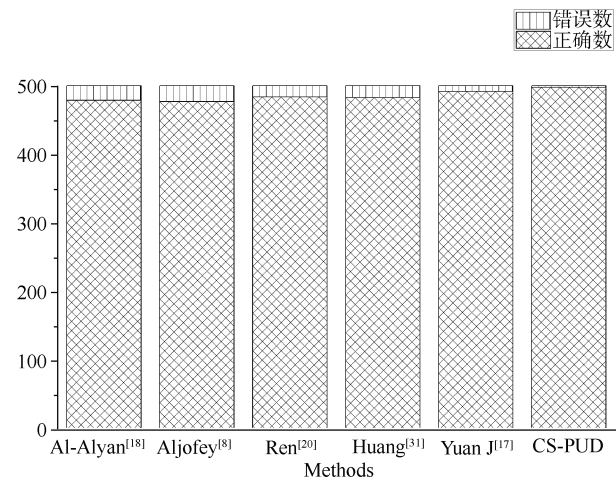


图 10 不同方法对 Openphish 未知数据的预测结果
Figure 10 Prediction results of different methods for unknown data from Openphish

基于以上分析, CS-PUD 方法通过在模型中引入胶囊网络以及对模型进行对抗训练提升了钓鱼 URL 检测模型预测未知数据的准确率, 也即泛化能力得到了提升。

(2) 具体案例分析

为进一步说明 CS-PUD 方法的有效性, 此处选取了 Phishtank 中最新被鉴定为钓鱼网站的 URL 作为分析样本, 如表 15 所示, 其中样本 1、2、3 在 Phishtank 社区中 ID 为 8109188、8109186、8109185, 均被鉴定为钓鱼网站, 是对样本 5 注册登录页面的伪造, 如图 11 所示。根据样本 1、2、3 与样本 4 的对比预测结果, 能够发现模型能够准确识别 URL 中“不正常”的字符序列, 并根据大量数据多次训练的结果检测出钓鱼 URL 的特征。根据样本 3、样本 4

以及样本 5 的预测结果, 能够发现模型对字母与数字的组合较为敏感, 并非针对 zimbra 这段字符。而样例 6 虽然模型预测为钓鱼网站的概率高达 40.63%, 但仍处于正常网站的范围之内。基于以上结果分析, 可见模型基本具备实际应用的能力。

表 15 具体案例检测分析
Table 15 Case specific testing analysis

编号	样本内容	网站类型	预测概率(%)
1	https://vn091-ab992.web.app	钓鱼网站	99.99
2	https://vn093-b4116.web.app	钓鱼网站	99.99
3	https://zimbranet365.web.app	钓鱼网站	99.99
4	https://web.app	正常网站	26.49
5	https://www.zimbra.com	正常网站	0.86
6	https://foxtool.web.app	正常网站	40.63

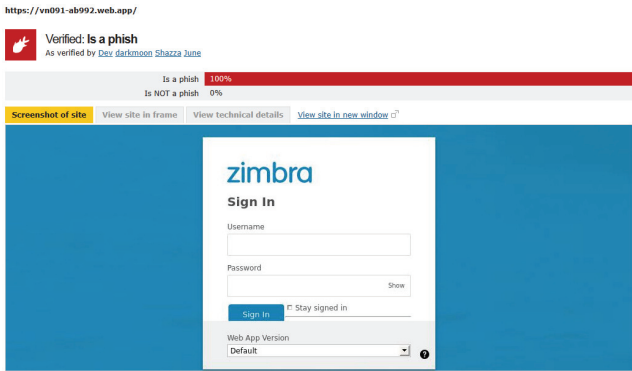


图 11 样本 1 的快照以及鉴定结果
Figure 11 Snapshot of sample 1 and identification results

5 结论

针对目前钓鱼网站检测研究在检测技术及特征提取方面陷入瓶颈, 同时基于深度学习的钓鱼网站 URL 检测方法存在模型泛化能力较弱, 测试集检测准确率明显低于训练集的情况, 对当前钓鱼 URL 检测在深度学习方面提取并利用 URL 字符序列特征的方式进行改进, 提出了一种具有更高检测准确率、更低误报率的检测方法。该方法首先对 URL 的不同组件进行分割得到 domain、path 以及 query 等组件, 使得模型对不同组件能够采取不同严格程度及更细粒度的特征判别。模型方面, 人为的将 URL 分为多个低维嵌入表示, 使每个低维特征进行线性变换, 最终将各组件的深层特征进行融合, 为避免卷积神经网络采用的池化策略过于关注局部特征而忽视整体空间结构, 对融合后的特征采用胶囊网络进一步提

① <https://openphish.com/feed.txt> 论文选取的数据为 2023.02.26 的数据

取, 得到考虑整体空间结构的深层特征。为提升模型的泛化能力, 独特的引入了对抗训练机制并对其进行改进以达到多嵌入层独立训练的目的, 增强了模型的泛化能力。在与其他同类方法的对比实验中, 结果表明本文所提出的基于 URL 组件分割的检测方法在准确率以及 F1-Score 上均有提升。

虽然 CS-PUD 方法对钓鱼 URL 的检测效果好, 但是随着攻击人员对钓鱼网站的精心掩饰, 钓鱼 URL 检测方法仍难以应对。后续本文会对基于组件分割的钓鱼网站检测方法进行拓展, 发挥 URL 快速检测的优点, 同时结合其他基于代码、主机信息等的检测方法共同对高造假水平的钓鱼网站进行检测。最后, 感谢本工作得到国家自然科学基金(No. 62272232), 江苏省自然科学基金(No. SBK2024041254)等项目的资助。

参考文献

- [1] APWG. Phishing Activity Trends Report, 3rd Quarter 2022. https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf December 12th, 2022.
- [2] Verma R, Dyer K. On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers[C]. *The 5th ACM Conference on Data and Application Security and Privacy*, 2015.
- [3] ENZE Y U, NURBOL, QING Y U. Phishing Website Detection Method Based on Integrated Learning[J]. *Computer Engineering and Applications*, 2019, 55(18): 81-88, 200.
(余恩泽, 努尔布力, 于清. 一种基于集成学习的钓鱼网站检测方法[J]. *计算机工程与应用*, 2019, 55(18): 81-88, 200.)
- [4] Chiew K L, Chang E H, Sze S N, et al. Utilisation of Website Logo for Phishing Detection[J]. *Computers & Security*, 2015, 54: 16-26.
- [5] Mohammad R M, Thabtah F, McCluskey L. Intelligent Rule-Based Phishing Websites Classification[J]. *IET Information Security*, 2014, 8(3): 153-160.
- [6] Sheng S, Wardman B, Warner G, et al. An Empirical Analysis of Phishing Blacklists[J]. *6th Conference on Email and Anti-Spam, CEAS 2009*, 2009.
- [7] Abdillah R, Shukur Z, Mohd M, et al. Phishing Classification Techniques: A Systematic Literature Review[J]. *IEEE Access*, 2022, 10: 41574-41591.
- [8] Aljofey A, Jiang Q S, Qu Q, et al. An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL[J]. *Electronics*, 2020, 9(9): 1514.
- [9] Zhang M, Xu B Y, Bai S, et al. A Deep Learning Method to Detect Web Attacks Using a Specially Designed CNN[M]. *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2017: 828-836.
- [10] Prakash P, Kumar M, Kompella R R, et al. PhishNet: Predictive Blacklisting to Detect Phishing Attacks[C]. *2010 Proceedings IEEE INFOCOM*, 2010: 1-5.
- [11] Sahoo D, Liu C H, Hoi S C H. Malicious URL Detection Using Machine Learning: A Survey[J]. *ArXiv e-Prints*, 2017: arXiv: 1701.07179.
- [12] Jain A K, Gupta B B. A Novel Approach to Protect Against Phishing Attacks at Client Side Using Auto-Updated White-List[J]. *EURASIP Journal on Information Security*, 2016, 2016(1): 9.
- [13] Sahingoz O K, Buber E, Demir O, et al. Machine Learning Based Phishing Detection from URLs[J]. *Expert Systems with Applications*, 2019, 117: 345-357.
- [14] Buber E, Diri B N, Sahingoz O K. NLP Based Phishing Attack Detection from URLs[M]. *Advances in Intelligent Systems and Computing*. Cham: Springer International Publishing, 2018: 608-618.
- [15] Daeef A Y, Ahmad R B, Yacob Y, et al. Wide Scope and Fast Websites Phishing Detection Using URLs Lexical Features[C]. *2016 3rd International Conference on Electronic Design*, 2016: 410-415.
- [16] Fu A Y, Liu W Y, Deng X T. Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)[J]. *IEEE Transactions on Dependable and Secure Computing*, 2006, 3(4): 301-311.
- [17] Yuan J T, Chen G X, Tian S W, et al. Malicious URL Detection Based on a Parallel Neural Joint Model[J]. *IEEE Access*, 2021, 9: 9464-9472.
- [18] AL-ALYAN A, AL-AHMADI S. Robust URL Phishing Detection Based on Deep Learning [J]. *KSII Transactions on Internet and Information Systems*, 2020, 14(7): 2752-68.
- [19] Yang P, Zhao G Z, Zeng P. Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning[J]. *IEEE Access*, 2020, 7: 15196-15209.
- [20] Bu S J, Cho S B. Deep Character-Level Anomaly Detection Based on a Convolutional Autoencoder for Zero-Day Phishing URL Detection[J]. *Electronics*, 2021, 10(12): 1492.
- [21] Zhao W, Ye J B, Yang M, et al. Investigating Capsule Networks with Dynamic Routing for Text Classification[EB/OL]. 2018: 1804.00538. <https://arxiv.org/abs/1804.00538v4>.
- [22] MIYATO T, DAI A M, GOODFELLOW I. Adversarial training methods for semi-supervised text classification [J]. arXiv preprint arXiv:160507725, 2016.
- [23] MADRY A, MAKELOV A, SCHMIDT L, et al. Towards deep learning models resistant to adversarial attacks [J]. arXiv preprint arXiv:1706.06083, 2017.
- [24] Jin W, Li Y X, Xu H, et al. Adversarial Attacks and Defenses on Graphs[J]. *ACM SIGKDD Explorations Newsletter*, 2021, 22(2): 19-34.
- [25] ZHU C, CHENG Y, GAN Z, et al. FreeLB: Enhanced Adversarial Training for Natural Language Understanding [J]. arXiv preprint arXiv:1909.11764, 2019.
- [26] Urcuqui C, Navarro A, Osorio J, et al. Machine Learning Classifiers to Detect Malicious Websites[J]. *SSN*, 2017, 1950: 14-17.
- [27] Mamun M S I, Ahmad Rathore M, Lashkari A H, et al. Detecting Malicious URLs Using Lexical Analysis[M]. *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2016: 467-482.
- [28] T. TIWARI, Phishing Site Urls(kaggle), <https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls> (2019).
- [29] Marchal S, François J, State R, et al. PhishStorm: Detecting Phish-

ing with Streaming Analytics[J]. *IEEE Transactions on Network and Service Management*, 2014, 11(4): 458-471.

- [30] Ren F L, Jiang Z W, Liu J. A Bi-Directional LSTM Model with Attention for Malicious URL Detection[C]. *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference*, 2019: 300-305.
- [31] Huang Y J, Yang Q P, Qin J H, et al. Phishing URL Detection via

CNN and Attention-Based Hierarchical RNN[C]. *2019 18th IEEE International Conference on Trust, Security and Privacy In Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering*, 2019: 112-119.

- [32] Safi A, Singh S. A Systematic Literature Review on Phishing Website Detection Techniques[J]. *Journal of King Saud University - Computer and Information Sciences*, 2023, 35(2): 590-611.



钟文康 于 2021 年在江西师范大学网络工程专业获得学士学位。现在南京理工大学网络空间安全专业攻读硕士学位。研究领域为 Web 安全、恶意检测领域。研究兴趣包括: 网络与 web 服务安全、恶意网站检测、恶意流量检测。Email: wenk@njust.edu.cn



张功萱 于 2004 年在南京理工大学计算机应用技术专业获得博士学位。现任南京理工大学教授、博士生导师。CCF 杰出会员、ACM 高级会员、IEEE 高级成员。研究领域为可信计算与计算机系统安全、网络与 web 服务安全。研究兴趣包括: 网络与 web 服务安全、多核和并行处理和分布式计算。Email: gongxuan@njust.edu.cn



王添 于 2022 年在南京理工大学网络空间安全专业获得博士学位。现任江苏财会职业学院信息工程学院讲师, 教研室主任, IEEE Member, CCF 会员, 江苏省计算机学会会员, 研究领域为云计算、边缘计算和网络物理系统。研究兴趣包括: 端-边-云资源管理与任务调度以及网络空间安全。迄今发表二十余篇相关领域高水平论文, 其中大部分发表于 IEEE TPDS, IEEE IoT, Elsevier JSA, IEEE HPCC, IEEE/ACM ASPDAC 等期刊和会议。Email: wangtian@jscfa.edu.cn