软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, [doi: 10.13328/j.cnki.jos.007119]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于双分支特征提取和自适应胶囊网络的 DGA 域名检测方法

杨宏宇[1,2], 章涛[2], 张良[3], 成翔[4], 胡泽[1]

[1](中国民航大学 安全科学与工程学院,天津 300300)

[2](中国民航大学 计算机科学与技术学院,天津 300300)

[3](亚利桑那大学 信息学院,图森,亚利桑那州,美国 AZ85721)

[4](扬州大学 信息工程学院,扬州 225127)

通讯作者: 胡泽, E-mail: zhu@cauc.edu.cn

**摘 要**: 面向域名生成算法(DGA, domain generation algorithm)的域名检测方法普遍具有特征提取能力弱、特征信息压缩比高等特点,这导致特征信息丢失、特征结构破坏以及域名检测效果较差等诸多不足.针对上述问题,提出一种基于双分支特征提取和自适应胶囊网络的 DGA 域名检测方法.首先,通过样本清洗和字典构建重构原始样本并生成重构样本集.其次,通过双分支特征提取网络处理重构样本,在其中利用切片金字塔网络提取域名局部特征,利用 Transformer 提取域名全局特征,并利用轻量级注意力融合不同层次的域名特征.然后,利用自适应胶囊网络计算域名特征图的重要度系数,将域名文本特征转换为向量域名特征,并通过特征转移计算基于文本特征的域名分类概率,同时利用多层感知机处理域名统计特征,以此计算基于统计特征的域名分类概率.最后,通过合并得到的两种不同视角的域名分类概率进行域名检测.大量的实验表明,本文所提方法在 DGA 域名检测以及 DGA 域名家族检测分类方面均取得了当前领先的检测效果,其中,在 DGA 域名检测中 F1 分数提升了 0.76%~5.57%,在 DGA 域名家族检测分类中 F1 分数(宏平均)提升了 1.79%~3.68%.

**关键词**: DGA 域名检测;深度学习;双分支特征提取网络;切片金字塔网络;自适应胶囊网络

**中图法分类号**: TP393

中文引用格式: 杨宏宇,章涛,张良,成翔,胡泽.基于双分支特征提取和自适应胶囊网络的 DGA 域名检测方法.软件学报. http://www.jos.org.cn/1000-9825/7119.htm/

英文引用格式: Yang HY, Zhang T, Zhang L, Cheng X, Hu Z. DGA domain name detection method based on double branch feature extraction and adaptive capsule network. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7119.htm

## DGA Domain Name Detection Method Based on Double Branch Feature Extraction and Adaptive Capsule Network

YANG Hong-Yu[1,2], ZHANG Tao[2], ZHANG Liang[3], CHENG Xiang[4], HU Ze[1]

[1](School of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China)

[2](School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

[3](School of Information, The University of Arizona, Arizona 85721, USA)

[4](School of Information Engineering, Yangzhou University, Yangzhou 225127, China)

**Abstract**: The existing domain name detection methods for domain generation algorithm (DGA) generally have the characteristics of weak feature extraction ability and high feature information compression ratio, which lead to feature information loss, feature structure destruction, and poor domain name detection performance. Aiming at the above problems, a DGA domain name detection method based on double branch feature extraction and adaptive capsule network is proposed. Firstly, the original samples are reconstructed through sample cleaning and dictionary construction, and the reconstructed sample set is generated. Secondly, the reconstructed samples are processed by

* 基金项目: 国家自然科学基金(62201576, U1833107); 中央高校基本科研业务费专项资金(3122022050); 中国民航大学信息安全测评中心开放基金(ISECCA-202202); 中国民航大学学科经费资助

收稿时间: 2023-09-10; 修改时间: 2023-10-30; 采用时间: 2023-12-15; jos 在线出版时间: 2024-01-05

a double branch feature extraction network, in which the domain name local features are extracted by using a sliced pyramid network, the domain name global features are extracted by using a transformer, and the features at different levels are fused by using lightweight attention. Then, using an adaptive capsule network to calculate the importance coefficient of the domain name feature map, convert domain name text features into vector domain name features, and calculate the domain name classification probability based on text features by feature transfer, meanwhile, using multilayer perceptron to process domain name statistical features, to calculate the domain name classification probability based on statistical features. Finally, domain name detection is performed by combining the domain name classification probabilities from two different perspectives. A large number of experiments show that the method proposed in this paper achieves leading detection results in DGA domain name detection and DGA domain name family detection and classification, where the F1-score in DGA domain name detection increased by 0.76% to 5.57%, and the F1-score(macro average) in DGA domain name family detection classification increased by 1.79% to 3.68%.

**Key words**: DGA domain name detection; deep learning; double branch feature extraction network; sliced pyramid network; adaptive capsule network

作为互联网的核心服务之一,域名系统(DNS, domain name system)实现域名和 IP 地址间的映射.然而 DNS 服务存在恶意域名检测精度低[1]、安全机制薄弱[2]等不足,导致网络攻击者滥用 DNS 发起恶意活动.僵尸网络[3]、恶意软件等恶意活动可以利用域名生成算法(DGA, domain generate algorithm)[4,5]批量生成用于建立通信渠道的候选域名,通过频繁切换域名以规避安全机制的拦截与屏蔽.大量 DGA 域名的出现给恶意域名检测带来了巨大挑战,因此研究并提出精准高效的 DGA 域名检测方法对保护网络安全至关重要.

目前的 DGA 域名检测方法主要分为基于黑名单和规则匹配的检测方法[6-7]、基于机器学习的检测方法[1,8,9]和基于深度学习的检测方法[10-12].基于黑名单和规则匹配的检测方法需要维护更新黑名单与规则库,存在开销大、及时性差等不足.并且目前基于深度学习的检测方法优于基于机器学习的检测方法.在检测过程方面,基于深度学习的方法可以自动挖掘域名特征,而基于机器学习的方法依赖特征工程,导致开销过大.在检测性能方面,基于深度学习的方法具有更优的性能[13].在最新的研究中,基于深度学习的方法的平均准确率比基于机器学习的方法的平均检测率高 1.22%[14].因此,基于深度学习的 DGA 域名检测方法已逐渐成为研究热点.

Vinayakumar 等人[15]从 DNS 日志中提取域名的网络层特征并使用多种深度学习模型检测 DGA 域名,但是检测精度均相对偏低.因为域名的网络层特征不是判断域名是否由 DGA 生成的决定因素.因此,使用网络层特征检测 DGA 域名可能导致检测精度降低.此外,从 DNS 日志中提取特征也会导致较高的时间成本.

Tran 等人[16]提出一种基于改进长短期记忆网络(LSTM, long short-term memory network)的 DGA 域名检测方法.该方法解决了类别数据不平衡问题,但对域名的字符信息和上下文时序信息等特征的利用率较低,导致对基于单词的 DGA 域名检测效果较差.Xu 等人[17]采用 *n*-gram 技术,提出一种基于字符的域名分类(*n*-CBDC, *n*-gram combined character based domain classification)检测方法.该方法将域名分割并提取域名特征,但仅提取了域名片段中的特征信息,没有考虑不同域名片段间的文本依赖关系,导致对基于单词的 DGA 域名检测精度较差.Yang 等人[18]提出一种基于轻量级全卷积网络和轻量级注意力(LFC-LA, lightweight full-convolutional network and lightweight attention)的 DGA 域名检测方法.然而,该方法存在特征丢失问题,导致检测精度降低.Namgung 等人[19]提出一种基于双向 LSTM 和 CNN 集成网络(BCEN, bidirectional LSTM-CNN ensemble network)的 DGA 域名检测方法.该方法利用 BiLSTM 和 CNN 分别提取域名的全局序列信息和局部序列信息,并直接拼接特征.该方法将全局和局部序列信息直接进行简单拼接,未充分考虑不同特征间的交互关系,并将拼接后的特征直接降维压缩用于域名检测,未充分挖掘语义信息,这些局限性均导致检测性能下降.Highnam 等人[20]提出一种基于 CNN 和 LSTM 的 Bilbo 模型用于检测 DGA 域名,通过 CNN 和 LSTM 分别提取域名的字符特征和时序信息.但该方法没有分析域名的语义特征,且没有融合提取的域名特征造成特征信息丢失,导致误报率较高.此外,CNN 和 LSTM 由于年代久远,其局部和全局特征提取能力较弱,已经无法跟上当前技术潮流.Tuan 等人[21]提出一种基于双向 LSTM 和注意力机制(BiLSTM-A, BiLSTM and attention)的 DGA 域名检测方法.但该方法无法感知不同域名间字符信息的差异,对基于随机字符的 DGA 域名检测效果较差.Huang 等人[22]提出一种基于具有预训练词嵌入的并行卷积神经网络(PEPC, parallel convolutional neural network with pre-trained

embeddings)的 DGA 域名检测方法.该方法可以感知不同范围内的字符信息,但存在特征信息丢失等缺陷,导致检测精度偏低.Liu 等人[23]提出一种基于 LSTM 和胶囊网络(LSTM-Caps, LSTM and capsule network)的 DGA 域名检测方法,但该方法没有分析域名字符特征且没有对胶囊信息进行优化,导致对基于随机字符的 DGA 域名检测效果不佳.

从上述文献分析可知,当前研究主要分为三类:(1) 使用单/双向 LSTM 及其变体提取全局特征进行 DGA 域名检测;(2) 使用 CNN 及其变体提取局部特征进行 DGA 域名检测;(3) 使用注意力机制或者直接拼接方式融合单/双向 LSTM 及其变体提取的全局特征和 CNN 及其变体提取的局部特征进行 DGA 域名检测.这三类研究主要具有如下局限性:(1) CNN 和 LSTM 及其简单变体均是比较陈旧的方法,特征提取能力已经无法满足当前技术潮流;(2) 特征融合机制过于简单,未进行进一步的语义特征深度挖掘就直接用于检测任务;(3) 域名检测任务依赖特征单一,仅考虑了文本特征,未引入不同视角的非文本统计特征辅助检测.这些局限性均导致特征信息丢失、特征结构破坏等诸多不足,从而严重影响了 DGA 域名检测性能.

针对上述不足,本文提出一种基于双分支特征提取和自适应胶囊网络的 DGA 域名检测方法.通过提取全局和局部两种不同层次的域名文本特征,并利用胶囊网络保护域名特征结构,同时引入域名统计特征辅助检测,有效地提高了检测精度.本文主要贡献如下:

(1) 提出一种切片金字塔网络(SPN, sliced pyramid network).通过域名切片方法和切片信息传递算法(SITA, slice information transmission algorithm)降低网络深度以减少信息丢失,提取有效的域名字符特征和上下文信息.

(2) 提出一种双分支特征提取网络(SPN-Former).并行提取不同层次的域名文本特征,并使用注意力机制实现不同特征间的信息交互,增强特征之间的关联性,以减少信息丢失,从而提高其特征提取能力.

(3) 提出一种自适应胶囊网络(ACN, adaptive capsule network).为增强域名特征,降低信息冗余度,引入通道注意力计算域名特征的特征图的重要度系数,并对特征图进行加权.为保护域名特征结构不被破坏,ACN 将标量域名特征转换为向量域名特征.为消除向量大小对检测结果的负面影响,通过改进动态路由算法,进一步提高 DGA 域名检测性能.

## 1 检测方法框架

目前,部分 DGA 域名检测方法通过分析 DNS 流量,提取域名的网络层特征进行域名检测.然而,攻击者可使用特定的规避方法避免产生异常的 DNS 流量[24],从而导致高漏报率.此外,DNS over HTTPS 解析机制[25]和 DNS 流量加密技术[26]已应用于互联网,DNS 信息的加密导致从 DNS 流量中提取特征的难度增加.因此,域名的网络层特征不适用于 DGA 域名检测.为提高检测性能,本文面向域名文本特征和统计特征,提出一种基于双分支特征提取和自适应胶囊网络的 DGA 域名检测方法.该方法框架如图 1 所示.

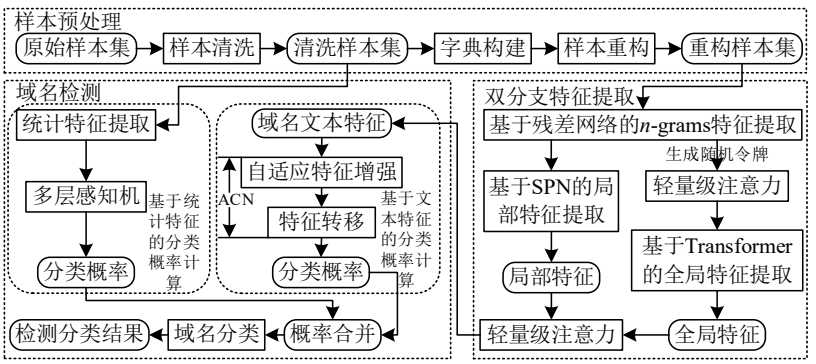



图 1 DGA 域名检测方法框架

基于双分支特征提取和自适应胶囊网络的 DGA 域名检测方法由样本预处理、双分支特征提取和域名检测 3 部分组成.各部分的功能设计如下:

(1) 样本预处理:对原始样本集进行样本清洗,得到清洗样本集,并创建域名字符到字符索引的映射字典,根据字典将字符串域名映射为数字化域名,得到重构样本集.

(2) 双分支特征提取:编码重构样本集中的数字化域名,提取域名的局部特征和全局特征.采用注意力机制实现局部特征与全局特征间的信息交互,丰富域名特征信息,从而得到域名文本特征.其中,局部特征包含域名的字符特征和上下文信息,全局特征包含域名的语义特征和上下文信息.

(3) 域名检测:首先,将域名文本特征输入自适应胶囊网络(ACN),通过自适应特征增强和特征转移方法计算基于文本特征的分类概率.其次,从清洗样本集中提取域名统计特征,并利用多层感知机计算基于统计特征的分类概率.最后,对基于文本特征的分类概率和基于统计特征的分类概率进行合并操作,并对域名进行分类,从而得到被检测域名的类型(是否为 DGA 域名)及家族.

## 2 双分支特征提取

### 2.1 样本预处理

在对被检测域名进行双分支特征提取前,首先进行样本预处理,包括样本清洗和样本重构 2 个过程.

在样本清洗过程中,删除重复域名及域名数量较少而难以分析的 DGA 域名家族,得到清洗样本集.

在样本重构过程中,将字符串域名映射为数字化域名,并将所有域名的长度设置为相同的大小.由于目前深度学习模型无法直接处理域名字符串等文本格式的数据,本文根据域名字符构建字符到字符索引的映射字典,并通过字典将每个字符映射为唯一的索引值,得到重构的域名样本.由于 SPN-Former 输入层接受固定长度域名,设置域名最大长度为 L 为 48[27],将域名长度大于 L 的域名进行截断处理,小于 L 的域名进行填充处理,从而得到重构样本集.当 L=15 时,样本重构过程如图 2 所示.

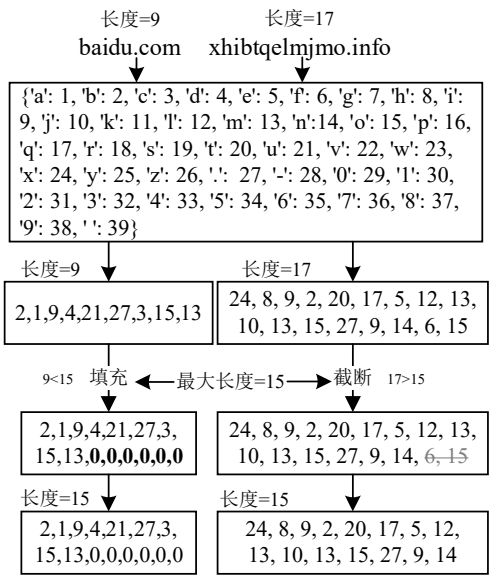



图 2 样本重构过程

### 2.2 SPN-Former网络架构

为提高检测方法的特征提取能力,减少特征信息丢失,本文提出一种基于改进的 Mobile-Former[28]的双分支特征提取网络(SPN-Former).

SPN-Former 的主要任务是并行处理域名的局部特征和全局特征,通过注意力机制实现不同特征间的信息交互,增强域名局部特征与全局特征的关联性,以减少特征信息丢失,提高特征提取能力.

SPN-Former 网络架构如图 3 所示.该网络由基于残差网络的 *n*-grams 特征提取、基于 SPN 的局部特征提取、全局令牌优化、基于 Transformer 的全局特征提取和特征优化 5 部分组成.

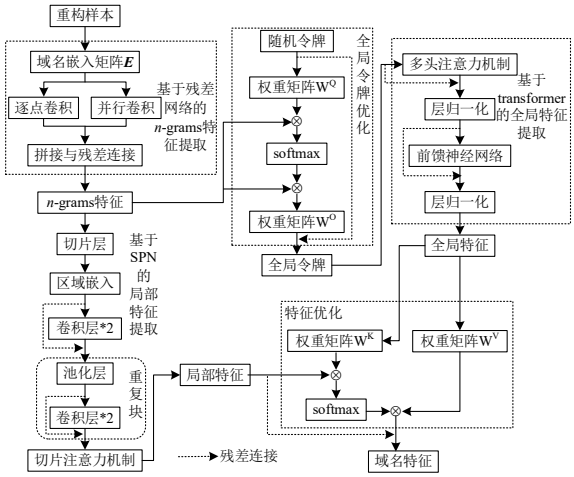



图 3 SPN-Former 网络架构

### 2.3 基于残差网络的*n*-grams特征提取

传统的 Mobile-Former 利用卷积[29]和瓶颈块[30]进行初步特征提取.但在域名检测中,通过卷积和瓶颈块无法提取域名的 *n*-grams 特征.因此,本文提出一种基于残差网络[31]的 *n*-grams 特征提取方法.该方法结合逐点卷积[32]和并行卷积[18],分别提取不同范围内的域名字符特征.基于残差网络的 *n*-grams 特征提取的过程如图 4 所示.

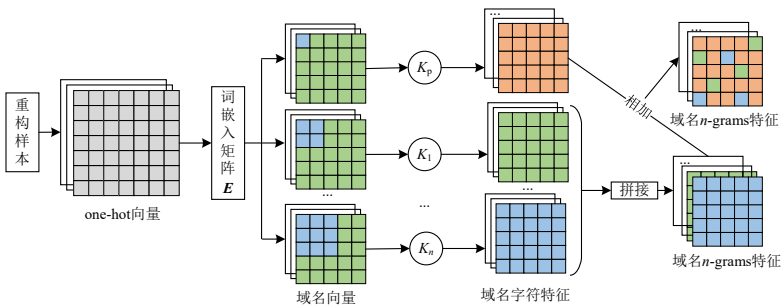



图 4 *n*-grams 特征提取过程

该方法由样本编码、*n*-grams 特征提取、*n*-grams 特征生成、特征升维和 *n*-grams 特征优化组成.具体过程设计如下:

(1) 样本编码:对重构样本进行 one-hot 编码,映射为二进制向量.但是 one-hot 向量存在数据稀疏且易导致特征空间过大等不足,因此通过词嵌入矩阵 $\boldsymbol{E}$ 将 one-hot 向量映射为分布式向量,从而得到域名词向量表示 $\boldsymbol{V}$.

假设组成域名的字符集大小为 $c$,域名最大长度为 $l$,则域名对应的 one-hot 向量 $\boldsymbol{M}\in\mathbb{R}^{l\times c}$.其中,向量 $\boldsymbol{M}$ 每一行表示一个字符的二进制向量,且对应字符位置的数值为 1,其他位置数值均为 0.但是任意两行间的向量距离均相等,导致域名的 one-hot 向量无法表达字符间的相似关系,且字符的词向量表示的维度大小只能为 $c$.词嵌入技术通过词嵌入矩阵 $\boldsymbol{E}$ 将 one-hot 向量映射为分布式向量的过程为

$$\boldsymbol{V}=\boldsymbol{M}*\boldsymbol{E} \tag{1}$$

其中,$\boldsymbol{E}\in\mathbb{R}^{c\times d}$,$d$ 为自定义设置的每个字符词向量表示的维度.

(2) $n$-grams 特征提取:构建并行卷积处理域名向量,提取域名的 $n$-grams 特征.并行卷积采用多个不同大小的卷积核,感知不同范围内的域名字符信息,得到多组不同的域名字符特征.并行卷积中的卷积操作为等长卷积,该卷积方式通过在两端补 0 以保持卷积后的域名特征向量大小不变,从而保护域名向量的边缘信息不丢失.并且,卷积后的域名特征向量的大小不变,便于向量计算.由第 $i$ 个卷积核得到的域名字符特征为

$$F_{\mathrm{c},i}=\boldsymbol{V}\otimes K_i \tag{2}$$

其中,$F_{\mathrm{c},i}$ 为通过第 $i$ 个卷积核获得的域名字符特征,$K_i$ 为第 $i$ 个卷积核,$\boldsymbol{V}$ 为样本编码后得到的域名词向量,⊗为卷积运算.

(3) $n$-grams 特征生成:线性拼接由并行卷积得到的多组域名字符特征,生成 $n$-grams 特征,生成 $n$-grams 特征的方法为

$$F_{\mathrm{n}}=[F_{\mathrm{c},i}]_{i=1:n} \tag{3}$$

其中,$F_{\mathrm{n}}$ 为域名 $n$-grams 特征,$F_{\mathrm{c},i}$ 为第 $i$ 个域名字符特征,$[\cdot]_{i=1:n}$ 为拼接操作,$n$ 为域名字符特征的数量,即并行卷积中卷积核的数量.

(4) 特征升维:通过逐点卷积,将域名向量的通道维度映射为与 $n$-grams 特征相同的通道维度.逐点卷积的卷积核大小为 1,且由卷积核的数量决定域名向量的通道数量.逐点卷积可在深度方向上对域名词向量进行加权组合,且不会改变每个通道的域名词向量的大小,可解决普通卷积造成的特征丢失问题.由逐点卷积得到的域名特征为

$$F_{\mathrm{p}}=\boldsymbol{V}\otimes K_{\mathrm{p}} \tag{4}$$

其中,$F_{\mathrm{p}}$ 为使用逐点卷积运算得到的域名特征,$K_{\mathrm{p}}$ 为逐点卷积的卷积核, $\boldsymbol{V}$ 为域名词向量,⊗为卷积运算.

(5) $n$-grams 特征优化:由于并行卷积操作会导致部分有价值的信息丢失,通过残差连接将由逐点卷积得到的域名特征,融合到域名的 $n$-grams 特征中,得到的新的 $n$-grams 特征为

$$F_{n-\mathrm{grams}}=F_{\mathrm{n}}+F_{p} \tag{5}$$

其中,$F_{n\text{-grams}}$ 为新的 $n$-grams 特征,$F_{\mathrm{n}}$ 为 $n$-grams 特征,$F_{\mathrm{p}}$ 为通过逐点卷积运算得到的域名特征.

### 2.4 基于SPN的局部特征提取

Mobile-Former 使用 MobileNet[30]提取局部特征.然而,MobileNet 无法提取域名的上下文信息.因此,为有效提取域名的局部特征,本文通过改进深度金字塔卷积神经网络[33],提出 SPN.采用 SPN 提取域名局部特征的过程如图 5 所示.具体方法设计如下:

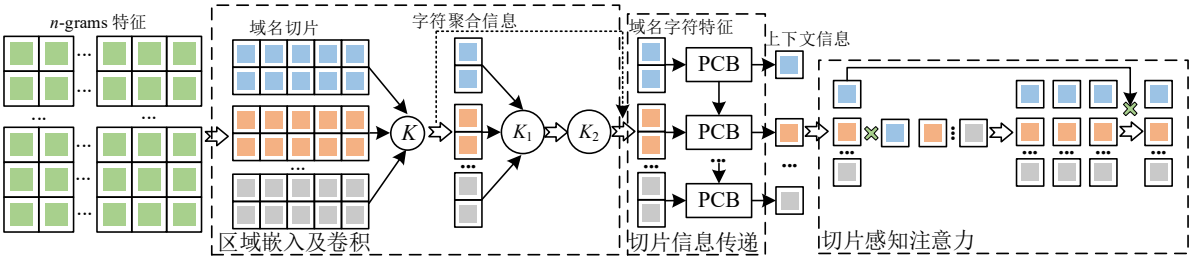



图 5 域名局部特征提取过程

(1) 为降低 SPN 网络的深度,保留更多有价值的域名特征信息,SPN 使用切片方法切分域名 $n$-grams 特征,得到若干大小相等的域名切片.通过切片方法将域名 $n$-grams 特征切分为 $F_{n\text{-grams}}=[F_{\mathrm{n},1}, F_{\mathrm{n},2}, \ldots, F_{\mathrm{n},i}, \ldots, F_{\mathrm{n},t}]$.

(2) 通过区域嵌入,提取每个域名切片的字符聚合信息.区域嵌入通过卷积核,对域名切片进行卷积操作.第

$i$ 个域名切片的字符聚合信息为

$$S_i = F_{\mathrm{n},i} \otimes K_{\mathrm{r}} \tag{6}$$

其中,$S_i$ 为第 $i$ 个域名切片的字符聚合信息,$F_{\mathrm{n},i}$ 为切分域名 $n$-grams 特征得到的第 $i$ 个域名切片,$K_{\mathrm{r}}$ 为区域嵌入使用的卷积核,⊗为卷积运算.

(3) 结合卷积和残差连接,从切片的字符聚合信息中进一步提取字符特征.其中卷积操作用以提取域名切片的字符特征,残差连接用于丰富字符特征信息以解决卷积过程中的信息丢失问题.提取第 $i$ 个域名切片的字符特征的方法为

$$F_{\mathrm{s},i} = S_i + \mathrm{ELU}(\mathrm{ELU}(S_i \otimes K_1) \otimes K_2) \tag{7}$$

其中,$F_{\mathrm{s},i}$ 为第 $i$ 个域名切片的字符特征,$S_i$ 为第 $i$ 个域名切片的字符聚合信息,$K_1$ 和 $K_2$ 分别为 2 个不同的卷积核,⊗为卷积运算,ELU[34]为非线性激活函数.

(4) 根据域名字符特征,使用 SPN 中的金字塔卷积块(PCB, pyramid convolutional block)进一步提取域名的上下文信息.为解决切片方法会破坏原域名的文本依赖关系,本文设计一种切片信息传递算法提取域名的上下文信息.切片信息传递算法如算法 1 所示.

**算法 1**.切片信息传递算法

输入:域名切片字符特征 $F_{\mathrm{s}}$;金字塔卷积块 PCB;

输出:域名切片上下文信息 $C$;

1. $c_0$ ← **0** //初始化零向量作为默认的切片上下文信息
2. $C$ ← [ ] //定义数组,存储切片上下文信息
3. $C$.append($c_0$) //将默认的上下文信息加入数组
4. **for** $i$ ← 1 to $len(F_{\mathrm{s}})$
5. $c'$ ← zeros_padding($C$[$i$-1]) //填充上下文信息
6. $s$ ← concatenate($F_{\mathrm{s}}$[$i$], $c'$) //融合上一个切片的上下文信息与当前的切片信息
7. $c$ ← PCB($s$) //使用 PCB 提取当前切片的上下文信息
8. $C$.append($c$) //将上下文信息加入数组
9. **end for**
10. $C$.remove($c_0$) //从数组中移除零向量
11. $C$=[$C$[$i$]]$_{i=1:len(C)}$ //拼接切片上下文信息
12. 结束算法返回域名切片上下文信息 $C$

(5) 构建切片注意力机制,感知不同域名切片间上下文信息的相关性,进一步提取关键的上下文信息并聚合所有域名切片的上下文信息得到域名局部特征.生成域名局部特征的计算过程为

$$e_{ij} = < c_i, c_j > \tag{8}$$

$$\varepsilon_{ij} = \mathrm{softmax}(e_{ij}) \tag{9}$$

$$m_i = \sum_{i=1}^{n} \varepsilon_{ij} c_j \tag{10}$$

$$X = [m_i]_{i=1:t} \tag{11}$$

其中,$X$ 为局域名局部特征,<·,·>为 2 个向量的点积,$c_i$ 为第 $i$ 个域名切片的上下文信息,$e_{ij}$ 为第 $i$ 个与第 $j$ 个域名切片的上下文信息的相似度,$\varepsilon_{ij}$ 为第 $i$ 个与第 $j$ 个域名切片的相关系数,$[\cdot]_{i=1:t}$ 为拼接 $t$ 个元素操作,$t$ 为域名切片的数量,softmax[35]为指数化归一函数.

此外,在域名局部特征的提取过程中,所有的卷积操作均不会改变域名特征图的维度,使域名中的所有字符映射到相同的向量空间,有利于合并相邻字符,从而提取到更多有价值的域名特征信息.

### 2.5 基于Transformer的全局特征提取

SPN-Former 通过全局令牌(固定维度的向量)[28]提取域名的全局特征.在提取被检测域名的全局特征前,需随机生成并优化全局令牌.全局令牌优化的具体方法设计如下:

(1) 向量切分:将全局令牌与域名的 *n*-grams 特征分别切片为若干长度和维度均相等的子向量.

(2) 令牌优化:使用轻量级多头注意力机制[28]将域名的 *n*-grams 特征信息传递到随机生成的全局令牌中以优化全局令牌.优化全局令牌的方法为

$$\boldsymbol{T}=[\mathrm{Atten}(\tilde{\boldsymbol{t}}_i\boldsymbol{W}_{Q\,i},\tilde{\boldsymbol{f}}_i,\tilde{\boldsymbol{f}}_i)]_{i=1:h}\boldsymbol{W}_O \tag{12}$$

其中,$\boldsymbol{T}$ 为优化后的全局令牌,$\tilde{\boldsymbol{t}}_i$ 为切分全局令牌后得到的第 $i$ 个全局令牌子向量,$\tilde{\boldsymbol{f}}_i$ 为切分 *n*-grams 特征后得到的第 $i$ 个 *n*-grams 特征子向量,$\boldsymbol{W}_{Q\,i}$ 为投影矩阵,$\boldsymbol{W}_O$ 为输出投影矩阵,Atten(·,·,·)为注意力运算,$[\cdot]_{i=1:h}$ 为拼接 $h$ 个元素操作.

优化全局令牌后,Transformer[36]通过全局令牌提取域名的全局特征.其中,Transformer主要由多头注意力机制和前馈神经网络(FNN, feed-forward neural network)组成.提取全局特征的具体方法设计如下:

(1) 通过多头注意力机制从全局令牌中学习并感知到区分不同类型或家族的域名的有效信息,其计算方法为

$$\boldsymbol{U}=\left[\mathrm{Atten}(\boldsymbol{Q}\boldsymbol{W}_{Q\,i},\boldsymbol{K}\boldsymbol{W}_{K\,i},\boldsymbol{V}\boldsymbol{W}_{V\,i})\right]_{i=1:h}\boldsymbol{W}_O \tag{13}$$

其中,$\boldsymbol{U}$ 为多头注意力机制提取的域名信息,$\boldsymbol{W}_{Q,i}$,$\boldsymbol{K}_{K,i}$,$\boldsymbol{V}_{V,i}$ 分别为第 $i$ 个注意力头中的 $\boldsymbol{Q}$,$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 对应的投影矩阵,$h$ 为多头注意力机制中的拆分的头的数量,$\boldsymbol{W}_O$ 为输出投影矩阵,$[\cdot]_{i=1:h}$ 为拼接 $h$ 个元素操作.在本文中,$\boldsymbol{Q}$,$\boldsymbol{K}$,$\boldsymbol{V}$ 均为全局令牌,Atten(·,·,·)为注意力运算.

(2) 使用 FNN 处理域名信息,进一步提取域名的语义信息和上下文信息,得到域名的全局特征.提取域名全局特征的方法为

$$\begin{aligned}&\boldsymbol{Z}_1=\mathrm{GELU}(\boldsymbol{W}_1\boldsymbol{U}+\boldsymbol{B}_1)\\&\quad\cdots\cdots\\&\boldsymbol{Z}_{L-1}=\mathrm{GELU}(\boldsymbol{W}_{L-1}\boldsymbol{Z}_{L-2}+\boldsymbol{B}_{L-1})\\&\boldsymbol{Z}=\mathrm{GELU}(\boldsymbol{W}_L\boldsymbol{U}_{L-1}+\boldsymbol{B}_L)\end{aligned} \tag{14}$$

其中,$L$ 为 FNN 的网络层数,$\boldsymbol{Z}_1$,$\boldsymbol{Z}_2$,..., $\boldsymbol{Z}_{L\text{-}1}$ 均为中间隐藏信息,$\boldsymbol{Z}$ 为全局特征,$\boldsymbol{W}_i$ 和 $\boldsymbol{B}_i$ 分别为 FNN 第 $i$ 层的权重矩阵和偏置矩阵,GELU[37]为非线性激活函数.

### 2.6 特征优化

通过轻量级交叉注意力机制[28]将域名的全局特征信息加权到域名局部特征的特征图中,对局部特征进行优化与提升.特征优化的具体方法设计如下:

(1) 切分:分别将域名局部特征和全局特征切分为若干相同大小的子向量.

(2) 特征增强:通过轻量级交叉注意力机制融合域名局部特征和全局特征,利用全局特征增强局部特征,增强后的域名局部特征为

$$\boldsymbol{F}=[\mathrm{Atten}(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{z}}_i\boldsymbol{W}_{K\,i},\tilde{\boldsymbol{z}}_i\boldsymbol{W}_{V\,i},)]_{i=1:h} \tag{15}$$

其中,$\boldsymbol{\Phi}$ 为增强后的域名特征,$\tilde{\boldsymbol{z}}_i$ 为域名全局特征切分后的第 $i$ 个子向量,$\tilde{\boldsymbol{x}}_i$ 为域名局部特征切分后的第 $i$ 个子向量,$\boldsymbol{W}_{K\,i}$ 为第 $i$ 个注意力头的键投影矩阵,$\boldsymbol{W}_{V\,i}$ 为第 $i$ 个注意力头的值投影矩阵,$[\cdot]_{i=1:h}$ 为拼接 $h$ 个元素操作,$h$ 为轻量级交叉注意力机制中头的数量, Atten(·,·,·)为注意力运算.

(3) 特征优化:使用增强后的特征对局部特征进行优化,得到具有全局特征信息和局部特征信息的域名文本特征,最终得到的域名文本特征为

$$\boldsymbol{L}=\boldsymbol{F}+\boldsymbol{X} \tag{16}$$

其中,$\boldsymbol{\Lambda}$ 为域名文本特征,$\boldsymbol{\Phi}$ 为增强后的域名特征,$\boldsymbol{X}$ 为 SPN 提取的域名局部特征.

## 3 域名检测

域名检测主要由基于文本特征的分类概率计算、基于统计特征的分类概率计算和概率合并 3 个部分构成.各部分的功能设计如下:

(1) 基于文本特征的分类概率计算:依据 SPN-Former 提取的域名文本特征,计算域名的分类概率.

(2) 基于统计特征的分类概率计算:从清洗样本集中提取与域名长度相关的统计特征,并依据统计特征计算域名的分类概率.

(3) 概率合并:合并基于文本特征的分类概率与基于统计特征的分类概率,检测域名类型或家族.

### 3.1 基于文本特征的分类概率计算

为防止域名文本特征结构被破坏并提高域名检测精度,本文对胶囊网络[38]进行改进,提出一种自适应胶囊网络(ACN)用于计算分类概率.ACN 网络架构如图 6 所示.ACN 由自适应特征增强、特征转移和分类概率计算 3 个部分构成.各部分的功能设计如下:

(1) 自适应特征增强:计算域名特征中特征图的重要度系数,赋予特征图不同的权重,以降低信息冗余度.将域名特征转换为初级胶囊,保护域名特征结构.

(2) 特征转移:通过动态路由算法分析初级胶囊间的耦合性,计算耦合系数,将保存信息相似的初级胶囊组合为高级胶囊.

(3) 分类概率计算:计算高级胶囊的长度,根据胶囊长度获取被检测域名的分类概率.

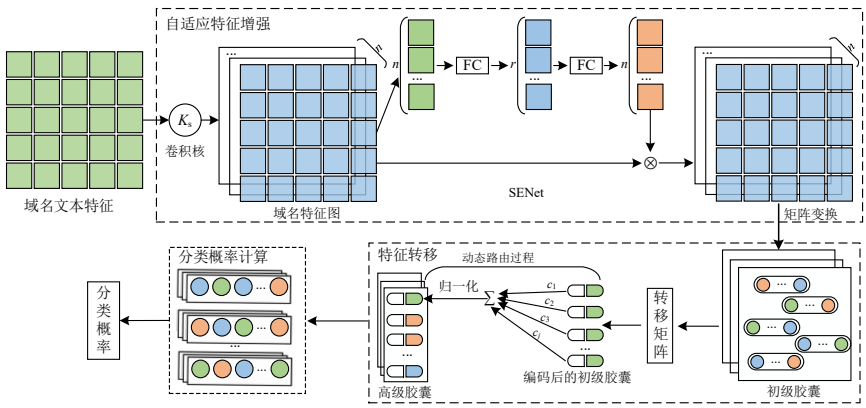



图 6 ACN 网络架构

#### 3.1.1 自适应特征增强

域名文本特征中一般包含多个特征图,但传统胶囊网络无法根据域名特征图的平均池化信息确定特征图的重要度,导致胶囊中保存的不重要的域名特征信息权重较大,信息冗余度高,检测网络无法充分利用有效信息.因此,本文设计了一种自适应特征增强方法.该方法由卷积、压缩-激励网络(SENet, squeeze-excitation networks)[39]和矩阵变换 3 个部分组成.具体方法设计如下:

(1) 卷积:对域名文本特征进行卷积操作,实现下采样.设置卷积核的大小为 3,步数的大小为 2[38].通过卷积操作得到的域名文本特征为

$$\boldsymbol{L}' = \boldsymbol{L} \otimes K_{\mathrm{s}} \tag{17}$$

其中,$\boldsymbol{\Lambda}'$为下采样后的域名文本特征,$\boldsymbol{\Lambda}$ 为双分支特征提取网络提取的域名文本特征,⊗为卷积运算,$K_{\mathrm{s}}$ 为卷积核.

(2) SENet:使用 SENet 优化域名特征,降低信息冗余度.首先计算每个特征图的平均池化信息,然后使用全连接网络处理平均池化信息并通过 sigmoid 函数计算每个特征图的重要度系数,赋予每个特征图不同的权重,以重点关注包含重要信息的特征图,从而降低信息冗余度.降低域名特征信息冗余度的计算过程为

$$q_i = \frac{1}{W \times H}\sum_{j=1}^{W}\sum_{t=1}^{H} M_i(j,t) \tag{18}$$

$$e_k = \mathrm{RELU}(b_k + \sum_{j=1}^{n} q_{kj} w_{jk}) \tag{19}$$

$$\alpha_i = \mathrm{sigmoid}(d_i + \sum_{j=1}^{r} e_{ij} u_{ji}) \tag{20}$$

$$\Lambda'' = [\alpha_i M_i]_{i=1:n} \tag{21}$$

其中,$M_i$ 为域名文本特征的第 $i$ 个特征图,$W$ 和 $H$ 分别为特征图的宽度与高度,$q_i$ 为特征图 $M_i$ 的平均池化信息,$e_i$ 为通过 SENet 压缩得到的中间隐藏信息,RELU[33]为非线性激活函数,$\alpha_i$ 为通过 SENet 和 sigmoid 函数[33]求得的第 $i$ 个特征图的重要度系数,$n$ 为特征图的数量,$r$ 为 SENet 压缩后的信息数量,$i \in [1,n]$,$k \in [1,r]$,$w_{jk}$ 和 $u_{ji}$ 分别是权重系数,$b_k$ 和 $d_i$ 分别是偏置系数,$[\cdot]_{i=1:n}$ 为拼接 $n$ 个元素操作,$\boldsymbol{\Lambda''}$为降低信息冗余度的域名文本特征.

(3) 矩阵变换:将标量域名特征转换为向量域名特征.标量域名特征以神经元形式存在,矩阵变换通过将多个神经元组合为一个初级胶囊,以初级胶囊保存特征信息.矩阵变换的具体过程如下:

1) 设置初级胶囊的维度为 $d$.

2) 根据每个域名文本特征向量中的神经元数量以及胶囊维度,计算转换每个特征需要的胶囊数量 $m$.

3) 按照形状$(m, d)$重新排列每个域名文本特征,得到以初级胶囊表示的域名文本特征.

4) 对每个胶囊进行归一化处理,将胶囊的长度映射到 0 到 1 之间.胶囊归一化处理的方法为

$$\mathrm{squash}(\boldsymbol{s}) = \frac{\| \boldsymbol{s} \|^2}{1+\| \boldsymbol{s} \|^2} \frac{\boldsymbol{s}}{\| \boldsymbol{s} \|} \tag{22}$$

其中,$\boldsymbol{s}$ 为胶囊表示的向量,$\|\cdot\|$为求模运算.

### 3.1.2 特征转移

特征转移是使用动态路由算法[38],将初级胶囊合并为高级胶囊.具体过程设计如下:

(1) 使用转移矩阵编码初级胶囊,将初级胶囊和高级胶囊投影到同一向量空间.第 $i$ 个编码后的初级胶囊为

$$\hat{\boldsymbol{u}}_i = \boldsymbol{W}_{ij}\boldsymbol{u}_i \tag{23}$$

其中,$\hat{\boldsymbol{u}}_i$ 为编码后的初级胶囊,$\boldsymbol{W}$ 为转移矩阵,$\boldsymbol{u}_i$ 为初级胶囊.

(2) 计算第 $i$ 个高级胶囊方法为:使用改进的动态路由算法分析胶囊之间的耦合性,计算耦合系数;根据耦合系数将所有的初级胶囊组合为一个高级胶囊;并计算该高级胶囊与每个初级胶囊间的相似度,并通过相似度更新耦合系数.当动态路由算法的循环结束,使用最后一组耦合系数计算最终的高级胶囊.动态路由算法如算法 2 所示.

**算法 2**.改进的动态路由算法

输入:编码后的初级胶囊 $\hat{\boldsymbol{u}}$;路由次数 $r$;

输出:高级胶囊 $\boldsymbol{v}_i$

1. $\boldsymbol{b}_{ij} \leftarrow 0$ ///初始化耦合系数
2. **for** $e \leftarrow 0$ to $r$-1
3. $ci \leftarrow softmax(\boldsymbol{b}_i)$ //耦合系数归一化
4. $\boldsymbol{s}_i \leftarrow \Sigma_j(c_{ij}\hat{\boldsymbol{u}}_j)$ //计算高级胶囊
5. $\boldsymbol{v}_i \leftarrow squash(\boldsymbol{s}_i)$ //高级胶囊归一化
6. $\boldsymbol{k}_1 \leftarrow \boldsymbol{v}_i / \| \boldsymbol{v}_i \|$ //计算 $\boldsymbol{v}_i$ 的单位向量
7. $\boldsymbol{k}_2 \leftarrow \hat{\boldsymbol{u}} / \| \hat{\boldsymbol{u}} \|$ //计算 $\hat{\boldsymbol{u}}$ 的单位向量
8. $\boldsymbol{b}_i \leftarrow \boldsymbol{k}_1 * \boldsymbol{k}_2 + \boldsymbol{b}_i$ //更新耦合系数
9. **end for**
10. $\boldsymbol{c}_i \leftarrow softmax(\boldsymbol{b}_i)$ //耦合系数归一化

11. $\boldsymbol{s}_i \leftarrow \Sigma_j(\boldsymbol{c}_{ij}\hat{\boldsymbol{u}}_j)$ //计算高级胶囊

12 $\boldsymbol{v}_i \leftarrow \text{squash}(\boldsymbol{s}_i)$ //高级胶囊归一化

13 结束算法返回高级胶囊 $\boldsymbol{v}_i$

3.1.3 分类概率计算

ACN 根据得到高级胶囊的长度计算域名的分类概率.具体过程设计如下:

(1) 胶囊长度计算:计算每个高级胶囊的长度.检测域名类型时,每个高级胶囊代表一种域名类型;检测域名家族时,每个高级胶囊代表一种域名家族;胶囊的长度代表域名是某种类型或属于某个家族的概率.每个胶囊长度的计算方法为

$$\|\boldsymbol{v}\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2} \tag{24}$$

其中,$\boldsymbol{v}$ 为高级胶囊,$v_i$ 是胶囊中的实例参数值,$n$ 为胶囊中实例参数的数量.

(2) 分类概率计算:由域名文本特征计算的第 $i$ 个域名的分类概率记为

$$R_i = (r_1, r_2, ... r_j, ..., r_N) \tag{25}$$

其中,$r_j$ 为胶囊长度,代表域名是第 $j$ 个类型或属于第 $j$ 个家族的概率,$N$ 为域名类型数量或域名家族的数量.

### 3.2 基于统计特征的分类概率计算

由于样本重构阶段将所有的域名长度统一为相同的长度,导致与域名长度相关的统计特征失效.为消除样本重构阶段带来的负面影响,在本文方法中,采用从清洗样本集中提取与域名长度相关的统计特征[40]的方法.提取的统计特征如表 1 所示.其中,熵表示域名字符排列的混乱程度,一般 DGA 域名的熵值高于合法域名的熵值,域名熵的计算方式如算法 3 所示.

**算法 3**.域名熵值计算算法

输入:域名字符串 s;

输出:域名熵值 entropy

```
1. cnts= {}    //初始化一个空字典
2. sum = len(s)    //sum 为域名长度
3. entropy = 0    // entropy 为域名熵值
4. for i ← 0 to sum
5.     cnts[s[i]] = cnts.get(s[i], 0) + 1    //统计每个字符出现的次数
6. end for
7. for i ← 0 to sum
8.     p = 1.0 * cnts.get(s[i]) / sum    //统计每个字符出现的频数
9.     entropy += -(p * math.log(p, 2))    //计算熵值
10. end for
11. 结束算法返回域名熵值 entropy.
```

**表 1** 域名统计特征

| 特征名称 | 说明 |
|---|---|
| 长度 | 域名本身的长度 |
| 特殊字符的频率 | 域名中特殊字符的数量 |
| 特殊字符比例 | 域名中特殊字符的占比 |
| 整数字符的频率 | 域名中整数字符的数量 |
| 整数字符比例 | 域名中整数字符的占比 |
| 元音字符的频率 | 域名中元音字符的数量(a, e, i, o, u) |
| 元音字符的比例 | 域名中元音字符的占比 |
| 熵 | 域名的混乱程度 |

提取域名的统计特征后,将域名表示为一个大小为 8 的向量 $\boldsymbol{X}$=($x_1$, $x_2$, …, $x_8$),然后通过多层感知机提取深

层的域名统计特征信息,使用 softmax 函数计算分类概率.具体过程如下:

(1) 深层统计特征信息提取:使用多层感知机将域名向量映射到不同的向量空间,聚合不同空间的特征信息.深层统计特征信息提取的处理过程为

$$\begin{aligned}&\boldsymbol{Z}_1=\mathrm{RELU}(\boldsymbol{W}_1\boldsymbol{X}+\boldsymbol{B}_1)\\&\cdots\cdots\\&\boldsymbol{Z}_{L-1}=\boldsymbol{W}_{L-1}\boldsymbol{Z}_{L-2}+\boldsymbol{B}_{L-1}\\&\boldsymbol{Z}=\boldsymbol{W}_L\boldsymbol{Z}_{L-1}+\boldsymbol{B}_L\end{aligned}\tag{26}$$

其中,$L$ 为多层感知机的网络层数,$\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, ..., $\boldsymbol{Z}_{L-1}$ 均为中间隐藏信息,$\boldsymbol{Z}$ 为多层感知机的输出,$\boldsymbol{W}_i$ 为第 $i$ 层的权重矩阵,$\boldsymbol{B}_i$ 为第 $i$ 层的偏置矩阵,RELU 为非线性激活函数.

(2) 分类概率计算:使用 softmax 函数处理多层感知机的输出计算分类概率,由统计特征计算的第 $i$ 个域名的分类概率记作

$$T_i=\mathrm{softmax}(\boldsymbol{Z})=(t_1,t_2,...t_j,...,t_N)\tag{27}$$

其中,$t_j$ 为域名是第 $j$ 个类型或属于第 $j$ 个家族的概率,$N$ 为域名类型数量或域名家族的数量.

### 3.3 域名检测分类

在基于域名文本特征和统计特征分别计算分类概率后,对得到的 2 个分类概率进行合并,然后对被检测域名的类型或家族进行检测分类.具体过程如下:

(1) 分类概率合并:比较域名属于相同类型或家族的概率,并取其中较大的概率值作为域名属于该类型或家族的概率,从而得到最终的分类概率.第 $i$ 个域名最终的分类概率为

$$\begin{aligned}&P_i=(p_1,p_2,...,p_j,...,p_N)\\&\text{其中,}\quad p_j=\begin{cases}r_j & if\ \ r_j>t_j\\ t_j & if\ \ r_j\le t_j\end{cases}\end{aligned}\tag{28}$$

其中,$r_j$ 和 $t_j$ 分别是由文本特征和统计特征计算的域名属于第 $j$ 个家族或类型的概率.

(2) 域名分类:比较域名的分类概率,概率值最大的域名类型或家族即为被检测域名的分类结果,第 $i$ 个域名的类型或家族为

$$\hat{y}_i=\mathrm{argmax}(P_i)\tag{29}$$

其中,$P_i$ 为第 $i$ 个域名的分类概率,在检测域名类型时,$\hat{y}_i\in[0,1]$,0 与 1 分别代表合法域名和 DGA 域名;在检测域名家族时,$\hat{y}_i\in\{z\in\mathbb{Z}|0\le z\le N\text{-}1\}$,$N$ 为域名家族的总数量,$j$ 代表第 $j$ 个域名家族.

## 4 实验结果与分析

### 4.1 实验设置

实验的计算机配置为:Intel(R) Xeon(R) Silver 4110 处理器,16GB 内存,NVIDIA Quadro P2000 GPU.训练和测试环境为 Windows 10,并且在实验中使用 CUDA 10.1 和 cuDNN 7.6 加速运算.实验代码使用 Python 3.7 和 TensorFlow 2.1 实现.

本文从 The Majestic Million 网站[41]发布的互联网访问量排名前十万的域名中收集了排名在五万内的域名,并从中随机采样取出部分域名构成合法样本集,同时从 360Netlab 数据集[42]中选择其中域名数量充足的域名家族并进行随机下采样获得DGA样本集,从而构成实验样本集.实验样本集的信息如表 2 所示,包括合法域名(benign)一共有 20 个域名家族.

**表 2** 实验样本集中各类域名的数量

| 域名家族 | 域名数量 | 域名家族 | 域名数量 | 域名家族 | 域名数量 | 域名家族 | 域名数量 |
|---|---|---|---|---|---|---|---|
| benign | 36154 | gameover | 3129 | pykspa-v1 | 11548 | simda | 7723 |
| banjori | 13571 | gozi | 4985 | ramnit | 5188 | suppobox | 4781 |
| bazardoor | 7352 | matsnu | 4994 | ranbyus | 4201 | tinba | 4987 |
| emotet | 2987 | murofet | 4305 | rovnix | 8756 | virut | 4877 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| flubot | 7796 | mydoom | 5014 | shiotob | 3536 | necurs | 4116 |

本文将实验样本集按 8:2 比例划分为训练集与测试集.在训练阶段,训练参数如表 3 所示.在测试阶段,采用常见的指标衡量本文方法的检测效果,分别为准确率、精确率、F1 分数、召回率、误报率、漏报率、ROC 曲线、曲线下面积(AUC, area under curve)、宏平均以及加权平均.

**表 3** 训练参数

| 参数 | 值 | 参数 | 值 | 参数 | 值 |
|---|---|---|---|---|---|
| 域名最大长度 | 48 | 全局令牌数量 | 3 | 学习率 | 1e-4 |
| 域名词向量维度 | 32 | 全局令牌维度 | 128 | 优算法 | Adam |
| 并行卷积核 | 2,3,5,7 | 初级胶囊维度 | 8 | 批大小 | 256 |
| SPN 通道大小 | 256 | 高级胶囊维度 | 16 | 训练轮次 | 200 |
| SPN 切片数量 | 4 | 路由次数 | 5 | 损失函数 | Focal Loss |

为验证本文方法的可行性与有效性,本文分别设置了 6 组验证实验,分别是(1) DGA 域名检测二分类实验;(2) DGA 域名家族检测分类实验;(3) 切片信息传递算法对检测性能的影响;(4) 域名最大长度 L 对检测性能的影响;(5) 域名统计特征对检测性能的影响; (6) 检测效率分析.

### 4.2 DGA域名检测二分类实验结果与分析

为验证本文方法的 DGA 域名检测性能,分别采用本文方法、LSTM[16]、LFC-LA[18]、Bilbo[20]、BiLSTM-A[21]和 PEPC[22]在本文的实验样本集上进行域名检测二分类实验.上述 6 种方法的检测结果如表 4 和图 7 所示,不同方法的 ROC 曲线如图 8 所示.

**表 4** DGA 域名二分类实验检测结果

| 方法 | 准确率 | F1 分数 | 精确率 |
|---|---|---|---|
| LFC-LA | 89.90% | 93.49% | 91.55% |
| Bilbo | 97.40% | 98.06% | 97.65% |
| BiLSTM-A | 97.42% | 98.30% | 98.41% |
| PEPC | 97.28% | 98.21% | 97.95% |
| LSTM | 97.34% | 98.24% | 98.41% |
| 本文方法 | **98.58%** | **99.06%** | **98.87%** |

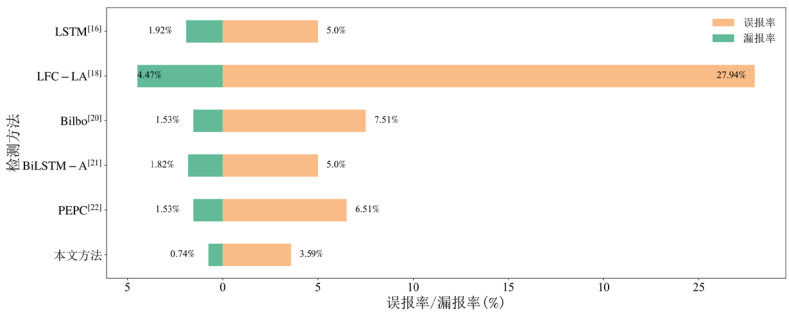



图 7 不同检测方法的误报率与漏报率

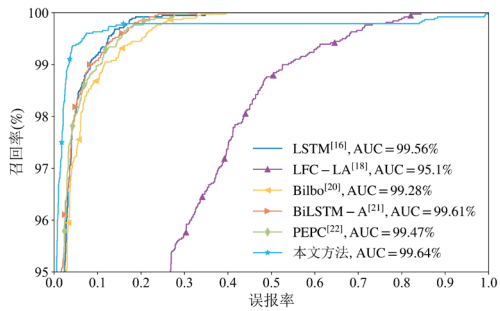


图 8　不同检测方法的 ROC 曲线

由表 4 可见,本文方法的准确率、F1 分数和精确率分别是 98.58%、99.06%和 98.87%.与其他 5 种方法相比,本文方法的准确率,F1 分数和精确率的最大提升幅度分别为 8.68%、5.57%和 7.32%,平均提升幅度分别为 2.71%、1.8%和 2.08%.

由图 7 可见,本文方法的误报率和漏报率分别是 3.59%和 0.74%.与其他 5 种方法相比,本文方法的误报率的最大降低幅度为 24.38%,平均降低幅度为 6.81%;本文方法的漏报率的最大降低幅度为 3.73%,平均降低幅度为 1.51%.

图 8 可见,本文方法的 AUC 为 99.64%.与其他 5 种方法相比,本文方法的 AUC 的最大提升幅度为 4.54%,平均提升幅度为 1.04%.

从上述对比结果可知,与其他 5 种主流方法相比,本文方法可以更加精准地检测 DGA 域名.其原因为:本文方法在使用域名字符特征和语义特征的基础上,结合域名上下文信息,对域名文本特征进行优化,减少域名特征信息的丢失.同时结合域名统计特征检测域名,保证特征信息的完整性,增强本文方法对 DGA 域名的检测能力.

为进一步验证本文方法的有效性,针对相同样本集,将本文方法分别与基于网络层特征的典型检测方法 LSTM[15]、循环神经网络(RNN, recurrent neural network)[15]和 Random Forest[15]进行检测对比,检测结果如表 5 所示.

**表 5**　与基于网络层特征的检测方法的检测性能对比

| 方法 | 准确率 | F1 分数 | 精确率 |
|---|---|---|---|
| LSTM | 97.6% | 93.8% | 89.2% |
| RNN | 96.5% | 90.6% | 83.8% |
| Random Forest | 94.6% | 85.1% | 76.2% |
| 本文方法 | **98.58%** | **99.06%** | **98.87%** |

由表 5 可见,与基于网络层特征的典型检测方法相比,本文方法的准确率、F1 分数和精确率均有明显提升.其中, 准确率、F1 分数和精确率的最大提升幅度分别为 3.98%、13.96%和 22.67%,平均提升幅度为 2.35%、9.23%和 15.8%.由此可知,通过域名文本特征和统计特征,可有效识别 DGA 域名.但通过网络层特征检测 DGA 域名得到的检测效果较差.因此,使用本文提取的域名文本特征和统计特征可以有效识别与检测 DGA 域名.

### 4.3 DGA域名家族检测分类实验与结果分析

为验证本文方法的 DGA 域名家族分类的检测效果,分别使用本文方法、LSTM[16]、*n*-CBDC[17]、BCEN[19]和 LSTM-Caps[23]进行 DGA 域名家族检测分类实验,在实验样本集上得到的 DGA 域名家族分类的 F1 分数、精确率和召回率分别如图 9 和表 6 所示.

由图 9 可见,本文方法在 DGA 域名家族检测分类实验中具有较好的分类效果,本文方法的宏平均 F1 分数和加权 F1 分数均高于其他对比方法.本文方法的宏平均 F1 分数和加权 F1 分数分别为 95.99%和 96.71%.与其他 4 种主流方法相比,宏平均 F1 分数和加权 F1 分数的最大提升幅度分别为 3.68%和 3.31%,平均提升幅度分别为 2.29%和 2.26%.

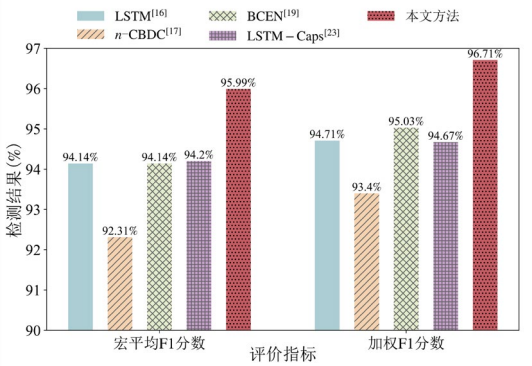



图 9 不同检测方法在 DGA 域名家族分类检测中的 F1 分数

**表 6** DGA 域名家族检测分类的精确率与召回率

| 方法 | LSTM | | *n*-CBDC | | BCEN | | LSTM-Caps | | 本文方法 | |
|---|---|---|---|---|---|---|---|---|---|---|
| 域名家族 | 精确率 | 召回率 | 召回率 | 召回率 | 精确率 | 召回率 | 精确率 | 召回率 | 精确率 | 召回率 |
| benign | 92.99% | 94.08% | 93.97% | 90.91% | 94.87% | 93.99% | 93.33% | 92.24% | **96.68%** | **97.16%** |
| banjori | 99.57% | 99.99% | 99.78% | 99.99% | 99.78% | 99.99% | 99.57% | 99.99% | **99.99%** | **99.99%** |
| bazardoor | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | **99.99%** | **99.99%** |
| emotet | 99.99% | 99.99% | 98.89% | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | **99.99%** | **99.99%** |
| flubot | 92.58% | 95.95% | 89.96% | 97.98% | 89.78% | **99.60%** | 91.63% | 97.57% | **93.05%** | 97.57% |
| gameover | 99.01% | **99.99%** | 99.99% | 97.00% | 98.04% | 99.99% | 99.00% | 99.00% | **99.99%** | 99.00% |
| gozi | 85.29% | 75.32% | 77.42% | 77.92% | 86.27% | 85.71% | 78.66% | 83.77% | **94.56%** | **90.26%** |
| matsnu | 91.08% | 86.67% | 84.21% | 87.27% | 91.14% | 87.27% | 88.96% | 87.88% | **94.61%** | **95.76%** |
| murofet | 89.40% | 87.66% | 85.26% | 86.36% | **99.16%** | 76.62% | 94.44% | 88.31% | 92.67% | **90.26%** |
| mydoom | **99.99%** | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | 99.99% | 99.47% | **99.99%** |
| necurs | 94.62% | **87.86%** | 96.77% | 64.29% | 96.58% | 80.71% | 94.40% | 84.29% | **96.80%** | 86.43% |
| pykspa-v1 | 97.89% | 99.99% | 98.82% | **99.99%** | 99.29% | 99.99% | 98.81% | 99.76% | **99.76%** | 99.76% |
| ramnit | 83.91% | 78.92% | 83.77% | 69.73% | 84.12% | 77.30% | **87.20%** | 77.30% | 84.88% | **78.92%** |
| ranbyus | **93.48%** | 89.58% | 84.18% | 92.36% | 89.26% | 92.36% | 91.03% | 91.67% | 92.41% | **93.06%** |
| rovnix | 99.67% | 99.01% | 98.37% | 99.34% | 99.34% | 99.99% | 99.99% | 99.34% | **99.99%** | **99.99%** |
| shiotob | **99.04%** | 95.37% | 98.06% | 93.52% | 92.04% | 96.30% | 98.11% | 96.30% | 97.20% | **96.30%** |
| simda | 96.75% | 96.75% | 95.72% | 99.99% | 98.01% | 99.99% | 98.37% | 98.37% | **98.80%** | **99.99%** |
| suppobox | 93.33% | 98.59% | 91.56% | 99.30% | 94.67% | 99.99% | 90.45% | 99.99% | **99.30%** | **99.99%** |
| tinba | 88.89% | 98.88% | 83.73% | 98.31% | 87.56% | 98.88% | 88.06% | **99.44%** | **91.15%** | 98.31% |
| virut | 93.01% | 92.36% | 89.38% | **99.31%** | 88.39% | 95.14% | **93.53%** | 90.28% | 93.10% | 93.75% |
| Macro | 94.52% | 93.85% | 92.49% | 92.68% | 94.41% | 94.19% | 94.28% | 94.28% | **96.22%** | **95.83%** |
| Weighted | 94.72% | 94.76% | 93.62% | 93.52% | 95.17% | 95.10% | 94.74% | 94.70% | **96.73%** | **96.74%** |

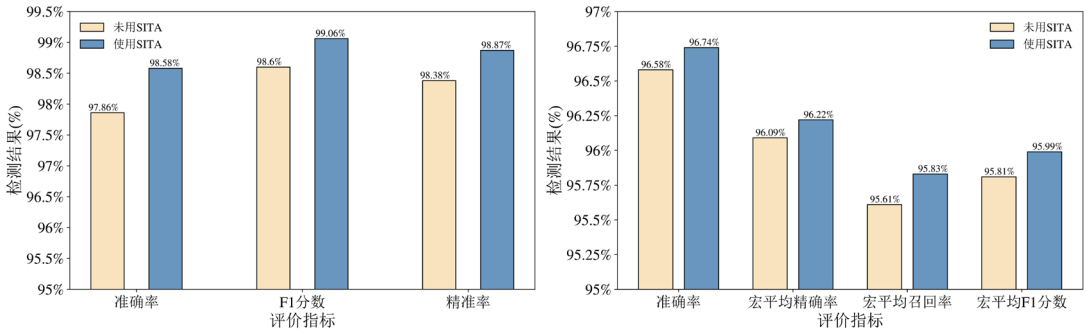
由表 6 可见,本文方法对合法域名及多数 DGA 家族的域名的检测精度均优于其他方法.其中,本文方法可以有效检测基于单词的 DGA 域名(gozi,matsnu,suppobox).其中,检测 gozi 家族域名时,精确率和召回率最大提升幅度为 17.14%和 14.94%,平均提升幅度为 12.65%和 9.58%;检测 matsnu 家族域名时,精确率和召回率最大提升幅度为 10.4%和 9.09%,平均提升幅度为 5.76%和 8.49%;检测 suppobox 家族域名时,精确率和召回率最大提升幅度为 8.85%和 1.4%,平均提升幅度为 6.8%和 0.52%.

从上述对比结果可知,与其他 4 种主流方法相比,本文方法在 DGA 域名家族检测分类方面具有更好的分类效果.其原因为:

(1) 在 DGA 域名家族分类实验中,LSTM、BCEN 和 *n*-CBDC 的特征提取能力较弱,仅提取某个层次的域名特征,没有充分利用其他的域名特征,导致特征信息丢失.此外,上述 3 种方法均会压缩域名特征,破坏域名特征结构,导致检测效果不佳.

(2) LSTM-Caps 没有破坏域名特征结构,但该方法没有对重要的特征图信息进行增强.此外,该方法仅考虑了域名的时序特征,存在特征信息丢失问题,从而导致对 DGA 域名家族分类的性能不佳.

(3) 本文方法同时处理域名局部特征和全局特征,实现了域名局部特征和全局特征间的交互,减少了域名特征信息丢失,提高了域名文本特征的质量.此外,本文方法通过 ACN 将标量域名文本特征转换为向量域名特征,同时分析域名文本特征中的特征图之间的相关性以优化域名文本特征,降低域名特征信息冗余度的同时保护了域名特征结构不被破坏.另外,还引入域名统计特征对域名进行分类,进一步提高了检测性能.




(a) 域名检测二分类检测结果 (b) 域名家族检测分类检测结果

图 10 SITA 对检测性能的影响

### 4.4 切片信息传递算法对检测性能的影响

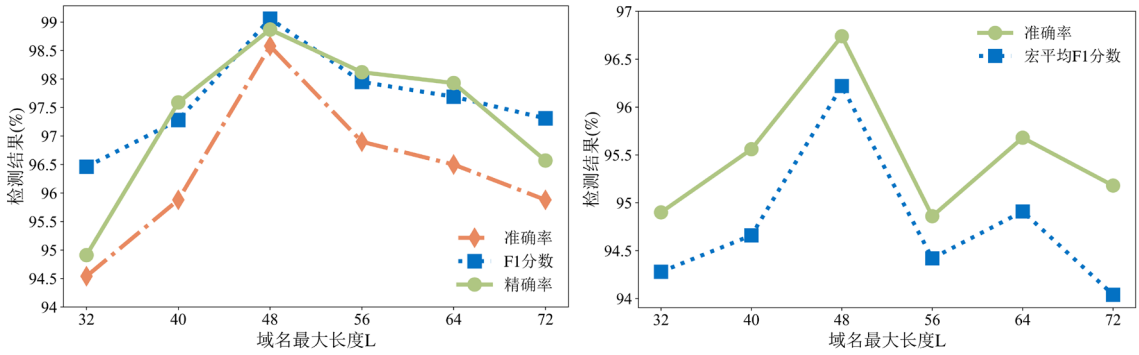
为验证切片信息传递算法对 DGA 域名检测分类的影响,本文分别在未使用以及使用切片信息传递算法时进行 DGA 域名检测分类实验,实验结果如图 10 所示.

由图 10(a)可见,通过使用 SITA,DGA 域名检测的检测性能有所提升,准确率、F1 分数和精确率分别提升了 0.72%、0.46%和 0.49%.由图 10(b)可见,DGA 域名家族分类的整体检测性能有明显提升,与不使用切片信息传递算法相比,其准确率、宏平均精确率、宏平均召回率和宏平均 F1 分数分别提升了 0.16%、0.13%、0.22%和 0.18%.其原因为:切片信息传递算法通过域名切片上下文信息的前向传播保证域名上下文信息的完整性,增强不同域名子序列间的关联.若仅使用切片方法,域名切片间的文本依赖关系会被破坏,从而导致检测结果不佳.

### 4.5 域名最大长度L对检测性能的影响

样本重构阶段会填充或截断域名样本,将所有域名设置为相同的长度 L.当 L 设置过大时,多数域名将被填充大量的无意义字符,给数据添加了噪声,影响检测性能;当 L 设置过小时,多数域名会被截断,导致 SPN-Former 无法提取到有效的域名文本特征.

为分析域名最大长度 L 对检测结果的影响,在本文研究中,将域名最大长度 L 设定为 32, 40, 48, 56, 64, 72, 并在实验样本集上进行 DGA 域名检测二分类实验和 DGA 家族检测分类实验,实验结果如图 11 所示.




(a) 域名检测二分类检测结果 (b) 域名家族检测分类检测结果

图 11 域名最大长度 L 对检测性能的影响

由图 11(a)可见,随着 L 的变化,本文方法的 DGA 域名检测性能呈现先上升后下降的趋势.当 L 为 48 时,本文方法取得最好的检测结果.

由图 11(b)可见,域名最大长度 L 对 DGA 域名家族分类检测结果影响较大.当 L 小于 48 时,随着 L 的增长,检测性能也逐步提升.当 L 为 48 时,检测性能达到最优.当 L 大于 48 时,随着 L 的增长,检测性能开始下降.当 L 为 64 时,检测性能有所提升,可能是样本集中大部分样本长度均小于 64,从填充部分提取的信息可映射为原样本长度的特征信息,从而提升了检测性能.但 L 为 64 时的检测性能不如 L 为 48 的检测性能.因此,本文设置最大域名长度为 48.

### 4.6 域名统计特征对检测性能的影响

样本重构会将所有域名长度设置为相同大小,导致原本域名的长度信息消失,降低域名长度对 DGA 域名检测的作用,同时与长度相关的统计特征也会失效.

为验证使用统计特征检测 DGA 域名的有效性,本文比较了使用域名文本特征和同时使用统计特征和文本特征进行 DGA 域名检测二分类实验和 DGA 家族检测分类实验的检测性能,实验检测结果分别如表 7 和图 12 所示.

**表 7** 域名统计特征对 DGA 域名检测的影响

| 特征 | 准确率 | F1 分数 | 精确率 |
|---|---|---|---|
| 文本特征 | 95.12% | 96.86% | 94.79% |
| 统计特征+文本特征 | 98.58% | 99.06% | 98.87% |

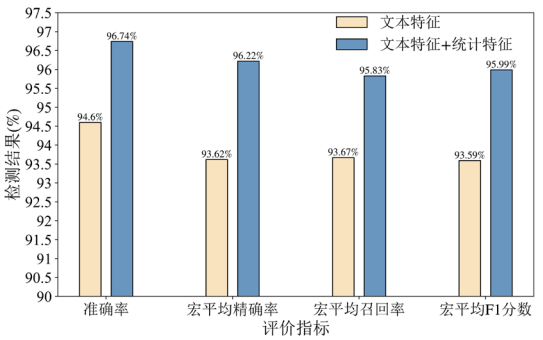



图 12 域名统计对 DGA 家族检测分类结果的影响

由表 7 可见,与仅利用文本特征检测 DGA 域名相比,本文方法的准确率、F1 分数和精确率分别提高了 3.46%,2.2%和 4.08%.

由图 12 可见,在添加统计特征进行 DGA 域名家族分类后,其准确率、宏平均精确率、宏平均召回率和宏平均 F1 分数分别提升了 2.14%、2.6%、2.16%和 2.4%.

由上述检测结果可知,本文方法通过引入域名本身长度的相关特征进行 DGA 域名检测和家族分类,能有效提高本文方法的检测性能,消除样本重构带来的负面影响.

### 4.7 检测效率分析

为评估检测效率,本文统计测试阶段所有方法推理时间所占的比例,其推理时间所占比例和检测性能如表 8 所示.

由表 8 可见,在 DGA 域名检测二分类实验和 DGA 域名家族检测分类实验中,本文方法所使用的推理时间所占比例优于其他大部分的方法.尽管本文方法所使用的模型比现有方法使用的模型复杂很多,但是在单位推理时间占比内,本文方法仍然取得了接近顶尖水平的显著性能.在不考虑时间性能的情况下,本文方法甚至取得了 SOTA 的检测性能.这些证据充分证明了本文提出的方法的优良特性,本文方法在检测性能和检测效率之间取得了平衡.

**表 8** 推理时间占比及检测性能

| 方法 | 推理时间占比 | 准确率 | 准确率/推理时间占比 |
|---|---|---|---|
| DGA 域名检测二分类 | | | |
| LSTM | 73.65% | 97.34% | 1.32 |
| LFC-LA | 76.26% | 89.90% | 1.18 |
| Bilbo | 72.13% | 97.40% | 1.35 |
| BiLSTM-A | 75.32% | 97.42% | 1.29 |
| PEPC | 70.54% | 97.28% | 1.38 |
| 本文方法 | 72% | 98.58% | 1.36 |
| DGA 域名家族检测分类 | | | |
| LSTM | 73.65% | 94.76% | 1.28 |
| *n*-CBDC | 71.21% | 93.52% | 1.31 |
| BCEN | 61.48% | 94.76% | 1.54 |
| LSTM-Caps | 85.58% | 94.70% | 1.11 |
| 本文方法 | 72% | 96.74% | 1.34 |

此外,本文通过使用 SPN 提取域名局部特征,不仅提高了域名特征的质量,还降低了域名特征表示的维度,减少了内存占用空间.

假设 SPN 中区域嵌入的卷积核大小为 $k$,PCB 的数量为 $m$,SPN 得到的切片数量为 $t$.域名的 $n$-grams 特征 $F_{n\text{-grams}}\in\mathbb{R}^{l\times d\times c}$,其中,$l$,$d$,$c$ 分别为域名长度,域名嵌入的维度和通道数.

SPN 利用切片层将 $F_{n\text{-grams}}$ 拆分为大小相同的子向量 $F_{\mathrm{n}}\in\mathbb{R}^{l/t\times d\times c}$,经 PCB 处理的数据序列的长度会减小为原来的一般,因此通过最后一层 PCB 处理后得到的向量 $\mathrm{X}\in\mathbb{R}^{l'\times d\times c}$,其中 $l'=(l/t-k+1)\times 2^{-m}$.

若使用传统的 DPCNN 提取域名特征,则经最后一层 PCB 处理得到的特征向量 $\mathrm{X}'\in\mathbb{R}^{l''\times d\times c}$,其中 $l''=(l-k+1)\times 2^{-m}$.

若使用其他 LSTM、LFC-A、PEPC 等方法提取域名特征,得到的域名特征 $\mathrm{X}''\in\mathbb{R}^{l'''\times d\times c}$,其中 $l'''=f(l)$且 $l'''$ 一般大于 $l\times 2^{-m}$.

由以上分析可知,相比其他方法,SPN 提取域名特征的过程中,使用的内存空间小于其他方法.

## 5 总结

针对现有 DGA 域名检测方法会导致域名特征信息丢失、域名特征结构被破坏且对 DGA 域名的检测精度较低等不足,提出一种基于双分支特征提取和自适应胶囊网络的 DGA 域名检测方法.通过双分支特征提取网络中的 SPN 提取域名局部特征,以保留更多有价值的域名信息;同时利用 Transformer 处理全局令牌得到域名全局特征.通过注意力机制实现域名局部特征与全局特征的信息交互,从而丰富特征信息,提高域名特征的质量.采用 ACN 降低域名特征信息冗余度并生成向量域名特征,保护了特征结构的完好性.由改进的动态路由算法处理向量域名特征生成高级胶囊以计算基于文本特征的分类概率;采用多层感知机处理域名统计特征以计算基于统计特征的分类概率.通过合并分类概率检测 DGA 域名.实验结果表明,本文方法对 DGA 域名的检测性能和对 DGA 域名家族分类的检测性能均优于现有主流方法.

在未来研究中,拟考虑在提取域名文本特征和统计特征的基础上引入并分析域名的 whois 信息,进一步提高 DGA 域名检测和 DGA 域名家族分类的检测性能.

**References**:

[1] Zhao H, Chen ZW, Yan RJ. Malicious domain names detection algorithm based on statistical features of URLs. In: Proc. of 2022 IEEE 25th Int'l Conf. on Computer Supported Cooperative Work in Design. Hangzhou: IEEE, 2022. 11-16.

[2] Cui J, Zhang L, Liu ZH, Li J, Shi L. An efficient framework for online malicious domain detection. In: Proc. of 11-th Int'l Congress on Image and Signal Processing, BioMedical Engineering and Informatics. Beijing: IEEE, 2018. 1-6.

[3] Zou FT, Tan Y, Wang L, Jiang YK. Botnet detection based on generative adversarial network. Journal on Communications, 2021, 42(7):95-106 (in Chinese with English abstract). https://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2021082 [doi: 10.11959/j.issn.1000-436x.2021082]

[4] Hoang XD, Vu XH. An improved model for detecting DGA botnets using random forest algorithm. Information Security Journal: A Global Perspective, 2022,31(4):441-450.

[5] Anderson HS, Woodbridge J, Filar B. DeepDGA: Adversarially-tuned domain generation and detection. In: Proc of the 2016 ACM Workshop on Artificial Intelligence and Security. New York: ACM, 2016. 13-21.

[6] Yoshida K, Fujiwara K, Sato A, Sannomiya S. Cardinality analysis to classify malicious domain names. In: Proc. of 2020 IEEE 44th Annual Computers, Software, and Applications Conference. Madrid: IEEE, 2020. 826-832.

[7] Chiba D, Yagi T, Akiyama M, Shibahara T, Yada T, Mori T, Goto S. DomainProfiler: Discovering domain names abused in future. In: Proc. of 2016 46-th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks. Toulouse: IEEE, 2016. 491-502.

[8] Bilge L, Sen S, Balzarotti D, Kirda E, Kruegel C. Exposure: A passive DNS analysis service to detect and report malicious domains. ACM Transactions on Information and System Security, 2014,16(4):1-28.

[9] Schüppen S, Teubert D, Herrmann P, Meyer U. FANCI: Feature-based automated NXDomain classification and intelligence. In: Proc. of 27th USENIX Security Symposium. Berkeley: USENIX Association, 2018. 1165-1181.

[10] Shahzad H, Sattar AR, Skandaraniyam J. DGA domain detection using deep learning. In: Proc. of 2021 IEEE 5-th Int'l Conf. on Cryptography, Security and Privacy. Zhuhai: IEEE, 2021. 139-143.

[11] Curtin RR, Gardner AB, Grzonkowski S, Kleymenov A, Mosquera A. Detecting DGA domains with recurrent neural networks and side information. In: Proc. of 2019 14-th Int'l Conf. on Availability. New York: ACM, 2019. 1-10.

[12] Pei XJ, Tian SW, Yu L, Wang HH, Peng YF. A two-stream network based on capsule networks and sliced recurrent neural networks for DGA botnet detection. Journal of Network and Systems Management, 2020,28:1694-1721.

[13] Ravi V, Alazab M, Srinivasan S, Arunachalam A, Soman KP. Adversarial defense: DGA-based botnets and DNS homographs detection through integrated deep learning. IEEE Transactions on Engineering Management, 2021, 70(1): 249-266.

[14] Hu XY, Chen H, Li M, Cheng G, Li RD, Wu H, Yuan YL. ReplaceDGA: BiLSTM based Adversarial DGA with High Anti-Detection Ability. IEEE Transactions on Information Forensics and Security, 2023,18: 4406-4421.

[15] Vinayakumar R, Soman K P, Poornachandran P. Detecting malicious domain names using deep learning approaches at scale. Journal of Intelligent & Fuzzy Systems, 2018,34(3):1355-1367.

[16] Tran D, Mac H, Van T, Tran AH, Giang NL. A LSTM based framework for handling multiclass imbalance in DGA botnet detection. Neurocomputing, 2018,275:2401-2413.

[17] Xu CY, Shen JZ, Du. Detection method of domain names generated by DGAs based on semantic representation and deep neural network. Computers & Security, 2019,85:77-88.

[18] Yang LH, Liu GJ, Wang JW, Bai HW, Zhai JT, Dai YW. Fast3DS: A real-time full-convolutional malicious domain name detection system. Journal of Information Security and Applications, 2021,61:102933-102946.

[19] Namgung J, Son S, Moon YS. Efficient deep learning models for DGA domain detection. Security and Communication Networks, 2021,2021:1-15.

[20] Highnam K, Puzio D, Luo S, Jennings NR. Real-time detection of dictionary DGA network traffic using deep learning. SN Computer Science, 2021,2(2):110-126.

[21] Tuan TA, Long HV, Taniar D. On detecting and classifying DGA botnets and their families. Computers & Security, 2022,113: 102549-102565.

[22] Huang WQ, Zong YY, Shi ZX, Wang LQ, Li PC. PEPC: A Deep parallel convolutional neural network model with pre-trained embeddings for DGA detection. In: Proc. of 2022 Int'l Joint Conf. on Neural Networks. Padua: IEEE, 2022. 1-8.

[23] Liu XY, Liu JM. DGA botnet detection method based on capsule network and *k*-means routing. Neural Computing and Applications, 2022,34(11):8803-8821.

[24] Liu LL. Research on DGA Detection Algorithm for IoT Botnet[MD. Thesis]. Xian:XIDIAN UNIVERSITY, 2023. (in Chinese with English abstract). [doi:10.27389/d.cnki.gxadu.2022.002288]

[25] Abu Al-Haija Q, Alohaly M, Odeh A. A Lightweight Double-Stage Scheme to Identify Malicious DNS over HTTPS Traffic Using a Hybrid Learning Approach. Sensors, 2023, 23(7): 3489.

[26] Lyu M, Gharakheili HH, Sivaraman V. A survey on DNS encryption: Current development, malware misuse, and inference techniques. ACM Computing Surveys, 2022,55(8):1-28.

[27] Liu XY, Liu JM, Liu C, Zhang YH. Novel botnet DGA domain detection method based on character level sliding windo ws and deep residual network. Acta Electronica Sinica, 2022, 50(1): 250-256. (in Chinese with English abstract). https://w ww.ejournal.org.cn/CN/Y2022/V50/I1/250 [10.12263/DZXB.20200619]

[28] Chen YP, Dai XY, Chen DD, Liu MC, Dong XY, Yuan L, Liu ZC. Mobile-former: Bridging mobilenet and transformer. In: Proc. of 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Louisiana: IEEE, 2022. 5270-5279.

[29] Albawi S, Mohammed T A, Al-Zawi S. Understanding of a convolutional neural network. In: Proc. of 2017 Int'l Conf. on Engineering and Technology. Antalya: IEEE, 2017. 1-6.

[30] Sandler M, Howard A, Zhu ML, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proc. of 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake: IEEE, 2018. 4510-4520.

[31] Zhang K, Sun M, Han TX, Yuan XF, Guo L, Liu T. Residual Networks of residual networks: multilevel residual networks. IEEE Trans on Circuits and Systems for Video Technology, 2017,28(6):1303-1314.

[32] Hua BS, Tran MK, Yeung SK. Pointwise convolutional neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. Salt Lake: IEEE, 2018. 984-993.

[33] Johnson R, Zhang T. Deep pyramid convolutional neural networks for text categorization. In: Proc. of 2017 55th Annual Meeting of the Association for Computational Linguistics. New York: ACL, 2017. 562-570.

[34] Rasamoelina AD, Adjailia F, Sinčák P. A review of activation function for artificial neural network. In: Proc. of 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herlany: IEEE, 2020. 281-286.

[35] Gao B, Pavel L. On the properties of the softmax function with application in game theory and reinforcement learning. arXiv preprint arXiv:1704.00805, 2017.

[36] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: Proc of the 31st Int'l Conf. on Neural Information Processing System. California: Curran Associates, 2017. 6000–6010.

[37] Hendrycks D, Gimpel K. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016: 1-11.

[38] Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. In: Proc of the 31st Int'l Conf. on Neural Information Processing System. California: Curran Associates, 2017. 1-11.

[39] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proc. of 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake: IEEE, 2018. 7132-7141.

[40] Park KH, Song HM, YOO JD, Hong SY, Cho B, Kim K, Kim, HK. Unsupervised malicious domain detection with less labeling effort. Computers & Security, 2022,116: 102662-102675.

[41] The Majestic Million: https://majestic.com/reports/majestic-million

[42] Zago M, Pérez M G, Pérez G M. UMUDGA: A dataset for profiling algorithmically generated domain names in botnet detection. Data in Brief, 2020, 30: 105400-105416.

**附中文参考文献**:

[3] 邹福泰,谭越,王林,蒋永康.基于生成对抗网络的僵尸网络检测.通信学报, 2021, 42(7): 95-106. https://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2021082 [doi: 10.11959/j.issn.1000-436x.2021082]

[24] 刘璐璐.面向物联网僵尸网络的 DGA 检测算法研究[D].西安:西安电子科技大学,2023.[doi:10.27389/d.cnki.gxadu.2022.002288]

[27] 刘小洋,刘加苗,刘超,张宜浩.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.电子学报, 2022, 50(1): 250-256. https://www.ejournal.org.cn/CN/Y2022/V50/I1/250 [10.12263/DZXB.20200619]