基于 BiLSTM-DAE 的多家族恶意域名检测算法

张 咪 彭建山
(河南建筑职业技术学院 河南 郑州 450064)
(数字工程与先进计算国家重点实验室 河南 郑州 450000)

摘 要 针对现有恶意域名检测算法对于家族恶意域名检测精度不高和实时性不强的问题,提出一种基于 BiLSTM-DAE 的恶意域名检测算法。通过利用双向长短时记忆神经网络(Bi-directional Long Short Term Memory, BiLSTM)提取域名字符组合的上下文序列特征,并结合深度自编码网络(Deep Auto-Encoder, DAE)逐层压缩感知提取类内有共性和类间有区分性的强字符构词特征并进行分类。实验结果表明,与当前主流恶意域名检测算法相比,该算法在保持检测开销较小的基础上,具有更高的检测精度。

关键词 恶意域名检测 深度自编码网络 双向长短时记忆神经网络 构词特征

中图分类号 TP309.5 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.10.047

MULTI-FAMILY MALICIOUS DOMAIN NAMES DETECTION ALGORITHM BASED ON BILSTM-DAE

Zhang Mi Peng Jianshan
(*Henan Technical College of Construction, Zhengzhou 450064, Henan, China*)
(*State Key Laboratory of Digital Engineering and Advanced Computing, Zhengzhou 450000, Henan, China*)

Abstract Aimed at the problem of poor detection accuracy and real-time performance of existing malicious domain name detection algorithms for family malicious domain names, a BiLSTM-DAE based malicious domain name detection algorithm is proposed. A Bi-directional long short term memory (BiLSTM) network was used to extract the context sequence features of domain name character, and deep auto-encoder (DAE) was used to extract and classify word formation features of strong characters layer by layer which were similarities within classes and distinctions between classes. The experimental results show that compared with the current mainstream malicious domain name detection algorithm, the algorithm has higher detection accuracy while keeping the detection overhead smaller.

Keywords Malicious domain names detection Deep auto-encoder Bi-directional long short term memory Word-formation features

0 引 言

由于域名系统(Domain Name System, DNS)自身缺少对攻击者威胁等恶意行为的检测能力,极易受到攻击者诱导点击欺诈、钓鱼网络等恶意攻击[1-3]。因此,如何快速、高效地检测出网络中的恶意域名攻击,是网络安全领域亟待解决的问题。

传统的恶意域名检测主要通过分析合法域名与恶意域名在字符组成上的差异[4],通常将恶意域名检测问题转换为字符相似度计算和字符串分类问题。如 Cucchiarelli 等[5]通过将域名分割为 2-gram 和 3-gram,并利用 KL(Kullback-Leibner)散度和 Jaccard 系数来计算合法域名与恶意域名的相似值。赵宏等[6]根据恶意域名黑名单与待测域名相似度计算值,构造了一种基于词法特征的恶意域名快速检测算法,取得了较好的

收稿日期:2021-04-28。2018 年度河南省科技攻关项目(182102310765)。**张咪**,讲师,主研领域:计算机应用。**彭建山**,副教授。

检测效果。张洋等[7]通过提取域名长度、分隔符个数、分隔符内字母与数字转换比例等特征,提出了一种基于多元属性的恶意域名检测方法。Truong 等[8]通过分析被动流量包解析数据中常见的 10 维特征构造指纹库,并计算待测域名流量包解析数据与指纹库的相似度值给出合法域名与恶意域名的判定。Fang 等[9]结合粗糙集增量式规则的机器学习分类算法,并利用 28 维域名字符特征构造分类模型。上述通过计算待测域名与提取的域名字符特征之间的相似度识别恶意域名是一种最直接的检测方法,但检测中大多依赖手工设计的特征,且特征维度有限,影响检测域名的种类和精度。

近年来,随着深度学习的快速发展,利用深度学习的相关理论与技术并结合域名的构造特征设计恶意域名检测算法成为主流方法[10-12]。如 Xu 等[13]结合深度学习和自然语言处理技术,构建了一种用于 DGA(Domains Generation Algorithm)生成的恶意域名检测模型,该模型针对特定类型的恶意域名具有较高的检测精度,但对多类域名字符串的拼接域名检测效果不佳。Almomani 等[14]提出了一种基于自适应进化神经网络的恶意域名检测算法,该类算法解决了多种字符拼接的恶意域名检测精度低的问题,但检测时间实时性不强。陈立皇等[15]采用 GRU(Gated Recurrent Unit)型循环神经网络并引入注意力机制,构建了一种 DGA 域名检测算法,有效解决了 DGA 生成的恶意域名随机性强而难以检测的问题,但仅针对特定家族的域名具有较高的性能。Sun 等[16]通过将实际网络场景中的客户端、域名、IP 地址等实体进行异构,提出了一种深度可扩展的异构图卷积网络的恶意域检测算法,该模型可以解决检测类型局限的问题,但仍存在实时性不强的缺点。

综上,针对当前恶意域名检测方法对多家族恶意域名检测精度不高和实时性不强的问题,提出了一种基于 BiLSTM-DAE 的多家族恶意域名检测算法。其中,BiLSTM 用于提取域名字符组合局部特征的上下文序列特征。DAE 在 BiLSTM 提取局部特征的基础上学习能够区分合法域名与恶意域名类内具有更近和类间具有更远距离的强特征。

1 BiLSTM-DAE 模型

1.1 模型组成

基于 BiLSTM-DAE 的恶意域名检测模型由输入层、特征提取层和输出层组成。网络结构如图 1 所示。

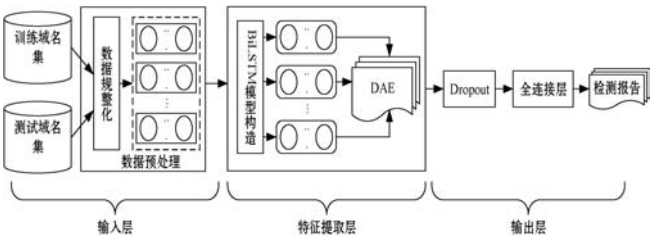


图 1 BiLSTM-DAE 网络结构

图 1 中,输入层采用 Keras 中的 Embedding 层将规整后的 URL(Uniform Resource Locator)映射为数值向量。特征提取层利用双向长短时记忆神经网络和深度自编码网络,学习区分合法域名与恶意域名类间有区分性和类内有共性的域名词法组成与结构等构词强特征。输出层将 DAE 的输出特征经过 Dropout 层和全连接层,给出分类结果。

1.2 输入层

通过收集与整理开源数据集和威胁检测报告中涉及的网络域名,并去除不完整域名,构造合法域名集和恶意域名集。本文采用二级、三级和四级等域名作为模型的输入,域名结构如图 2 所示。

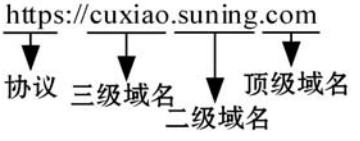


图 2 URL 结构

首先对中的每一字符进行向量化。现有字符向量化主要包括 One-hot 编码表示和分布式编码表示等两种,其中 One-hot 编码表示如[0, 1, …, 0, 0, 0],其中 1 表示域名字符串中某字符在字典集中的位置,当字典集维度较大时,此编码方式计算复杂度较大,且编码分布稀疏,难以表达上下文语义信息。分布式编码表示因其出色的上下文信息包含能力被广泛地应用[17],维度可控,可以将字符映射为固定长度的向量。因此,本文选用分布式编码对 URL 中的每一字符进行向量化,如图 3 所示。URL 中每一字符量化前需统计域名 URL 中常出现的字符。本文根据 ASCII 码字符集设定映射字典为 128 维。

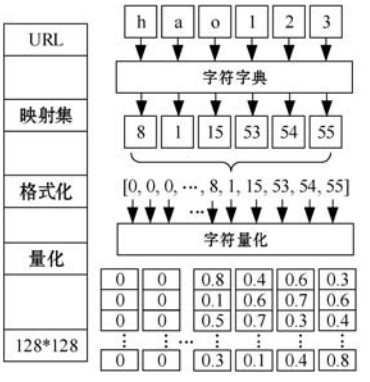


图 3 域名向量化

图 3 以“hao123”为例说明域名量化过程。首先将输入域名字符串统一为定长的 L,当域名字符串长度大于 L 时,对超出部分进行裁剪;当域名字符串长度不足 L 时,采用0 补齐,本文 L 取值 128。具体裁剪采用式(1)。

$$s(\boldsymbol{url}_i)=\begin{cases}\boldsymbol{Ze}+\boldsymbol{url}_i & len(\boldsymbol{url}_i)<L\\ \boldsymbol{url}_i & len(\boldsymbol{url}_i)=L\\ \boldsymbol{url}_i[0,L-1] & len(\boldsymbol{url}_i)>L\end{cases}\tag{1}$$

式中:$\boldsymbol{s}(\boldsymbol{url}_i)$表示经过裁剪后的定长向量,$\boldsymbol{url}_i$ 表示每一标准化的域名;$\boldsymbol{Ze}$ 表示零向量。

1.3　特征提取层

(1) BiLSTM 模型。

合法域名和恶意域名在构造规则和字符组合等方面相对自由,但在字符与字符组合上仍存在上下文依赖关系。因此,本文利用图 4 中的双向长短时记忆神经网络 BiLSTM 提取域名的上下文特征。

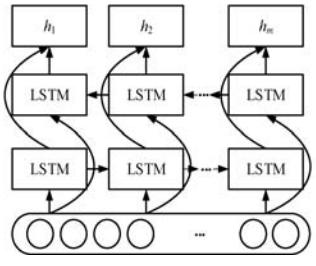


图 4　BiLSTM 结构

BiLSTM 由正向 LSTM 和反向 LSTM 结合而成,假设给定一个包含 m 个字符的域名 $Domain=\{d_1\ d_2\cdots\ d_m\}$ 按 d_1 到 d_m 的顺序输入至 BiLSTM 模型学习类间有区分性且类内有共性的特征。特征表示如式(2)所示。

$$Fo=\{\overrightarrow{LSTM}(\vec{\boldsymbol{h}}_{t-1},e_i)(\overleftarrow{\boldsymbol{h}}_{t-1},e_i)\}\tag{2}$$

式中:$\vec{\boldsymbol{h}}_{t-1}=\overrightarrow{LSTM}(\vec{\boldsymbol{h}}_{t-2},e_{i-1})$表示输入为 e_{i-1}时正向 LSTM 隐藏层的输出,$\overleftarrow{\boldsymbol{h}}_{t-1}=\overleftarrow{LSTM}(\overleftarrow{\boldsymbol{h}}_{t-2},e_{i-1})$表示输入为 e_{i-1}时反向 LSTM 隐藏层的输出。

(2) DAE 模型。

BiLSTM 层的输出特征虽保留了上下文语义信息,但特征冗余,使得模型分类时间开销增加,因此本文利用自编码网络通过对输入数据的编码压缩,并根据压缩的特征恢复压缩前原始输入数据降低特征维度,获得区分合法域名与恶意域名的类间有区分性和类内有共性的强特征。自编码网络的结构如图 5 所示。

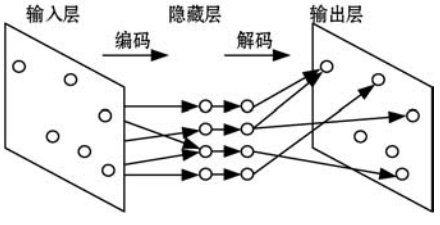


图 5　自编码网络结构

自编码网络输入端与输出端的神经元个数相同,极易导致输出端直接复制输入端,因此引入稀疏性限制。计算如式(3)所示。

$$\begin{cases}\hat{\rho}_j=\dfrac{1}{m}\sum\limits_{i=1}^{m}[k_j(x^i)]\\ KL(\rho\parallel\hat{\rho}_j)=\rho\log\dfrac{\rho}{\hat{\rho}_j}+(1-\rho)\log\dfrac{1-\rho}{1-\hat{\rho}_j}\\ Jsparse(x,\hat{x})=J(x,\hat{x})+\theta\sum\limits_{j=1}^{m}KL(\rho\parallel\hat{\rho}_j)\end{cases}\tag{3}$$

式中:k_j表示第 j 层隐藏层神经元相对于输入 x 的激活值;θ 为稀疏性限制项的权重;$\hat{\rho}_j$ 表示第 j 层隐藏层神经元的平均激活系数;ρ 为稀疏性因子;$KL(\rho\parallel\hat{\rho}_j)$为稀疏性限制项。

由于单层自编码网络 AE 通过压缩感知学习区分合法域名与恶意域名类间有区分性且类内有共性的强特征的能力有限,因此,本文通过叠加多个单层自编码网络,构造深度自编码网络,并采用逐层贪婪训练算法对每层网络进行训练,即将 BiLSTM 层的输出特征作为深度自编码网络单层自编码器的输入,将域名的分布式表征作为自编码网络的输入逐层压缩感知,直至获取能够区分合法域名与恶意域名的强特征为止。图 6 给出了深度自编码网络的结构图,其中最后一层连接具有分类识别功能的分类器。

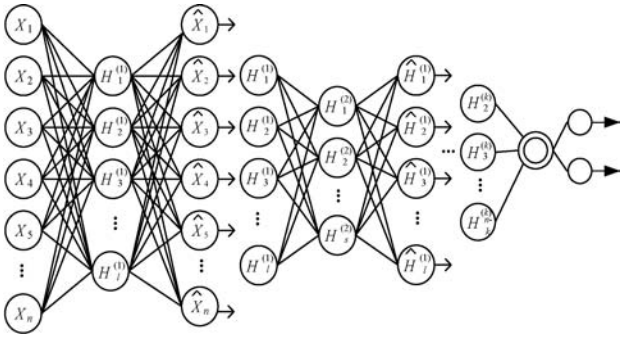


图 6　深度自编码网络结构

图 6 所示的深度自编码网络按照逐层压缩感知的方式进行训练,学习区分合法域名与恶意域名的强特征。其中,$F_K=\{X_1,X_2,\cdots,X_n\}$,$n$ 为 BiLSTM 的输入特征集容量大小。训练步骤如下:

步骤 1　按照单层降噪自编码网络的训练方式进行训练,得到第一层网络的初始参数,然后将第一层网络的输出作为第二层网络的输入。

步骤 2　第二层网络以同样的方式进行训练,得到该层的网络参数,然后将第二层网络的输出作为第三层网络的输入,以此类推,对网络进行逐层训练。

步骤 3　网络最后一层为分类器,以最后压缩得到的数据作为输入,对输入域名进行分类,并根据分类结果误差来调整各层自编码网络。

1.4 输出层

将压缩感知学习的区分合法域名与恶意域名的强特征进行拼接,得到特征向量集合 $\boldsymbol{H}$。特征向量表示如式(4)所示。

$$\boldsymbol{H}=\{h_1,h_2,\cdots,h_k\} \tag{4}$$

将 $\boldsymbol{H}$ 输入到一个具有分类识别功能的多重感知器 Softmax 中,利用式(5)计算获得输入域名的分类值。

$$MO_i=\text{Softmax}(\boldsymbol{w}_i\boldsymbol{d}_{it}+b_i) \tag{5}$$

式中:MO_i 为输出域名类别;w_i 为全连接层与输出层之间的权重矩阵;d_{it} 为 t 时刻全连接层的输出矩阵,b_i 为偏置。

2 实验与结果分析

本文实验选用深度学习框架 PyTorch 1.8 和深度学习库 Keras,Intel Core i7 R9000P,GPU 为 RTX3060 16 GB,使用 Python 语言搭建模型。

2.1 数据集与评价标准

实验数据集中合法域名来源于开源网站 Alexa 和安全联盟,随机选择 260 000 条域名构造合法域名集;恶意域名从 360Netlab、DNSBL(DNS Black List Query System)、MDL(Malware Domain List)中整理恶意域名 86 000 条,构造恶意域名集,其中 360Netlab 包含 26 种家族恶意域名。此外,将合法域名样本集和恶意域名样本集混合后按照 8∶1∶1的比例划分为训练集、测试集和验证集,域名数据集详细信息如表 1 所示。

表 1 域名数据集

类型		描述	数量
合法域名集		Alexa、安全联盟	260 000
恶意域名集	360Netlab	Gameover、Blackhole、Symmi、Dircrypt、Bamital、Tinba、Necurs、Ramdo、Ranbyus、Rovnix、Emotet、Corebot、Qakbot、Kraken、Locky、Pykspa、Simda、Virut、Chinad、Dyre、Shifu、Shiotob、Suppobox、Banjori、Cryptolocker、Rammit	51 000
	Malware Domain List	http://www. malwaredomainlist. com/	23 000
	DNS Black List Query System	https://dnsbl. abuse. ch/	12 000

采用准确率 A_{ccuracy}、精确率 P_{recision}、假阳性率(False Positive Rate, F_{PR})和假阴性率(False Negative Rate, F_{NR})作为评价标准,计算如式(6)所示。

$$\begin{cases} A_{\text{ccuracy}}=\dfrac{T_p+T_n}{T_p+T_n+F_p+F_n} \\ P_{\text{recision}}=\dfrac{T_p}{T_p+F_p} \\ F_{PR}=\dfrac{F_p}{T_n+F_p} \\ F_{NR}=\dfrac{F_n}{T_n+F_n} \end{cases} \tag{6}$$

式中:T_p 表示被正确判定为合法域名的个数;F_n 表示被误报为合法域名的个数;F_p 表示被误报为恶意域名的个数;T_n 表示被正确判定为恶意域名个数。

2.2 参数设置

本文模型采用参数固定的方法确定初始值,模型初始化时所需要确定的参数如表 2 所示。此外,为防止模型过拟合,在 BiLSTM 层后加入 Dropout 层,设定 Dropout 率为 0.5。

表 2 模型超参数设置

参数指标	值
词向量维度	128
优化函数	Adam
学习率	0.001
BiLSTM 隐藏层节点	128
批次大小	18
丢弃率	0.5

2.3 实验结果与分析

基于 BiLSTM-DAE 的恶意域名检测算法对 Malware Domain List 和 DNS Black List Query System 列表中的恶意域名的检测性能如表 3 所示。对 360Netlab 中包含的 26 种家族恶意域名的检测性能如表 4 所示。由表 3 和表 4 可知,本文算法在 DNSBL、MDL 和 360Netlab 等三个数据集上的检测性能表现良好,究其原因是本文算法通过利用 BiLSTM 和 DAE 结合的串行混合模型 BiLSTM-DAE 在学习域名上下文语义信息的基础上,能够学习区分合法域名与恶意域名的类间有区分性和类内有共性的强特征,利用学习到的强特征较好地完成待测域名的分类。

表 3 DNSBL 和 MDL 检测性能(%)

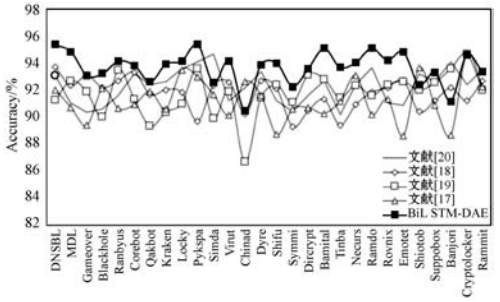
算法	A_{ccuracy}	P_{recision}	F_{PR}	F_{NR}
DNSBL	96.38	95.47	4.26	4.11
MDL	96.85	96.44	4.09	3.92

表 4　26 种家族恶意域名检测性能(%)

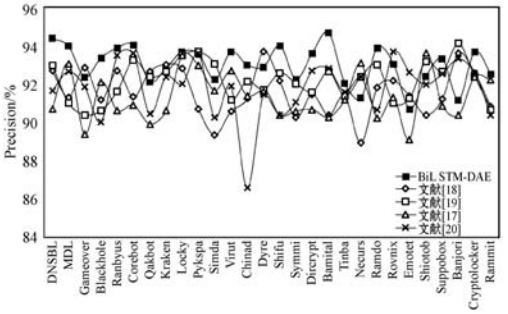
类型	$A_{ccuracy}$	$P_{recision}$	F_{PR}	F_{NR}	类型	$A_{ccuracy}$	$P_{recision}$	F_{PR}	F_{NR}
Gameover	96.80	94.04	4.11	5.28	Symmi	95.86	96.41	4.66	4.04
Blackhole	96.80	97.18	3.20	3.09	Dircrypt	96.37	97.65	4.23	3.58
Ranbyus	91.06	90.42	8.39	8.54	Bamital	95.02	96.38	4.36	4.22
Corebot	97.45	97.11	2.22	2.24	Tinba	97.19	96.52	3.09	3.16
Qakbot	97.34	98.53	2.26	2.08	Necurs	90.28	90.10	9.64	9.79
Kraken	90.81	88.06	9.02	9.44	Ramdo	96.11	94.35	3.14	3.60
Locky	93.56	94.84	7.22	6.96	Rovnix	97.01	97.22	3.19	3.15
Pykspa	90.37	92.10	9.49	9.17	Emotet	94.54	96.03	5.34	4.92
Simda	95.88	94.06	4.23	4.09	Shiotob	94.69	93.82	4.75	4.91
Virut	92.44	93.61	7.31	7.28	Suppobox	97.16	97.02	2.26	2.30
Chinad	97.02	96.83	3.17	3.11	Banjori	94.39	92.14	5.15	5.60
Dyre	97.64	97.90	3.10	3.08	Cryptolocker	94.89	95.13	5.74	5.26
Shifu	96.51	95.88	3.50	3.61	Rammit	97.31	96.59	2.87	3.01

2.4　同类相关工作对比

在相同的实验环境和评价标准下,分别构造文献[17]基于可分离卷积的轻量级恶意域名检测模型、文献[18]基于 LSTM 与多头注意力机制的恶意域名检测算法、文献[19]基于自编码神经网络的恶意域名检测算法、文献[20]基于 LSTM 和 CNN 的恶意域名检测方法和本文 BiLSTM-DAE 的恶意域名检测算法。在不同数据集上各模型的性能比较如图 7 所示,其中图 7(a)为 5 种模型在 360Netlab(26 种家族恶意域名集)DNSBL 和 MDL 等数据集上的 Accuracy 对比结果。图 7(b)为 5 种模型在 360Netlab(26 种家族恶意域名集)DNSBL 和 MDL 等数据集上的 Precision 对比结果。表 5 给出了不同模型在相同测试集上的时间开销对比结果。



(a) 在 Accuracy 上的性能对比



(b) 在 Precision 上的性能对比

图 7　不同数据集上各模型的性能比较

由图 7 可知,本文模型在 Accuracy 和 Precision 等方面优势明显,主要原因是单一 LSTM、CNN、DAE 只考虑了部分上下文特征或局部强特征。虽然 LSTM-CNN 模型考虑了上下文信息和局部强特征,但本文 BiLSTM-DAE 串行混合模型在充分考虑域名上下文双向依赖关系的基础上,深层次地学习能够区分合法域名与恶意域名类内有共性和类间有区分性的强特征,能够更好地区分合法域名与恶意域名,实验也验证了本文模型的高效性。

表 5　不同模型的时间开销对比

对比指标	文献[17]	文献[18]	文献[19]	文献[20]	BiLSTM-DAE
检测总数/条	28 019	26 564	26 362	30 083	32 069
时间开销/s	34	31	29	38	36
速率(条·s^{-1})	824.08	856.90	909.03	791.66	890.81

由表 5 可知,在时间开销方面,文献[17]、文献[18]、文献[19]和文献[20]等模型的检测速率分别为 824.08、856.90、909.03、791.66;本文 BiLSTM-DAE 模型在 36 s 内完成 32 069 条域名的准确识别,检测速率为 890.81 条/s,虽本文模型在时间总开销上相比文献[17]、文献[18]和文献[19]有所增加,但在单位时间内与文献[17]、文献[18]和文献[20]相比,检测速率优势明显。此外,由图 7 可知,在检测精度方面,本文模型相比文献[17]、文献[18]、文献[19]和文献[20]等模型分别提升了 2.67 %、1.90 %、1.93 %、1.54 %,可在最高 $A_{ccuracy}$、$P_{recision}$和最低 F_{NR}和 F_{PR}下达到 890.81 条/s 的检测速率,验证了本文模型在保持检测时间开销较小的基础上,具有较高的检测精度。

3 结 语

本文提出了一种基于 BiLSTM-DAE 的恶意域名检测算法,通过学习域名上下文语义信息的基础上,深层次地学习了区分合法域名与恶意域名类间有区分性和类内有共性的字符特征。在不同数据集上进行测试,结果表明本文 BiLSTM-DAE 的混合模型在保持检测时间开销较小的基础上具有较高的检测精度,可有效提高实际网络场景中对僵尸网络、垃圾邮件等的检测能力。

参 考 文 献

[1] Weaver R. Visualizing and modeling the scanning behavior of the Conficker botnet in the presence of user and network activity[J]. IEEE Transactions on Information Forensics and Security,2015,10(5):1039 - 1051.

[2] 常兆斌. 基于域名构词特征的分阶段恶意域名检测算法研究[D]. 兰州:兰州理工大学,2020.

[3] 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.

[4] Zhu J L, Peng G H, Wang D W. Dual-domain-based adversarial defense with conditional VAE and Bayesian network [J]. IEEE Transactions on Industrial Informatics,2021,17(1):596 - 605.

[5] Cucchiarelli A, Morbidoni C, Spalazzi L, et al. Algorithmically generated malicious domain names detection based on n-grams features[J]. Expert Systems with Applications, 2020,170:114551.

[6] 赵宏,常兆斌,王乐. 基于词法特征的恶意域名快速检测算法[J]. 计算机应用,2019,39(1):227 - 231.

[7] 张洋,柳厅文,沙泓州,等. 基于多元属性特征的恶意域名检测[J]. 计算机应用,2016,36(4):941 - 944,984.

[8] Truong D, Tran D T, Huynh B. Detecting malicious fast-flux domains using feature-based classification techniques[J]. Journal of Internet Technology,2020,21(4):1061 - 1072.

[9] Fang L M, Yun X Y, Yin C, et al. ANCS: Automatic NX-domain classification system based on incremental fuzzy rough sets machine learning[J]. IEEE Transactions on Fuzzy Systems,2020,29(4):742 - 756.

[10] 王志强,李舒豪,池亚平,等. 基于深度学习的恶意 DGA 域名检测[J]. 计算机工程与设计,2021,42(3):601 - 606.

[11] Yang L H, Zhai J T, Liu W, et al. Detecting word-based algorithmically generated domains using semantic analysis[J]. Symmetry,2019,11(2):1 - 20.

[12] 吴警,芦天亮,杜彦辉. 基于 Char-RNN 改进模型的恶意域名训练数据生成技术[J]. 信息网络安全,2020,20(9):6 - 11.

[13] Xu C Y, Shen J Z, Du X. Detection method of domain names generated by DGAs based on semantic representation and deep neural network[J]. Computers and Security,2019, 85:77 - 88.

[14] Almomani A, Nawasrah A, Alauthman M, et al. Botnet detection used fast-flux technique based on adaptive dynamic evolving spiking neural network algorithm[J]. International Journal of Ad Hoc and Ubiquitous Computing,2021,36(1): 50 - 65.

[15] 陈立皇,程华,房一泉. 基于注意力机制的 DGA 域名检测算法[J]. 华东理工大学学报(自然科学版),2019,45(3):478 - 485.

[16] Sun X Q, Wang Z L, Yang J H, et al. Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks[J]. Computers and Security,2020,99: 102067.

[17] 杨路辉,白惠文,刘光杰,等. 基于可分离卷积的轻量级恶意域名检测模型[J]. 网络与信息安全学报,2020,6(6): 112 - 120.

[18] 唐永旺,刘欣. 基于 Bi-LSTM 和自注意力的恶意代码检测方法[J]. 计算机应用与软件,2021,38(3):327 - 333.

[19] 丁红卫,万良,龙廷艳. 深度自编码网络在入侵检测中的应用研究[J]. 哈尔滨工业大学学报,2019,51(5):185 - 194.

[20] 张斌,廖仁杰. 基于 CNN 与 LSTM 相结合的恶意域名检测模型[J]. 电子与信息学报,2021,43(10):2944 - 2951.

(上接第 268 页)

[18] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS 2017.

[19] Wang L, Li S, Lü Y, et al. Learning to rank semantic coherence for topic segmentation[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. EMNLP, 2017.

[20] LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series[J]. The Handbook of Brain Theory and Neural Networks,1995,3361(10):1995.

[21] Chen H, Branavan S R K, Barzilay R, et al. Global models of document structure using latent permutations[C]//Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. ACL, 2009.

[22] Arnold S, Schneider R, Cudré-Mauroux P, et al. Sector: A neural model for coherent topic segmentation and classification[J]. Transactions of the Association for Computational Linguistics,2019,7:169 - 184.

[23] Utiyama M, Isahara H. A statistical model for domain-independent text segmentation[C]//Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. ACL, 2001.