E-mail:xblxb@ sdu.edu.cn
http://lxbwk.njournal.sdu.edu.cn
Tel:+86-0531-88366917

文章编号:1671-9352(2019)07-0106-07 DOI:10. 6040/j.issn.1671-9352.0.2019.249

一种基于深度学习的快速 DGA 域名分类算法

刘洋[1],赵科军[1,2*],葛连升[1],刘恒[3]

(1.山东大学信息化工作办公室, 山东 济南 250100; 2.山东大学计算机科学与技术学院, 山东 青岛 266237; 3.中电长城网际系统应用有限公司, 北京 102209)

摘要:提出了一种基于深度学习的 CNN-LSTM-Concat 快速 DGA 域名分类算法,使用多层一维卷积网络对域名字符进行序列化处理,LSTM 网络层用于强化获取字符间长距离依赖关系。通过将 LSTM 的多序列输入转化为单向量输入,在保证检测性能的前提下,能够大幅提高训练和检测速度。实验证明,我们的方法对 DGA 域名分类的准率在公开数据集上达到 98.32%。同时,在准确率相比主流的 LSTM 方法更高的情况下,检测时间比 LSTM 方法快 6.41 倍。

关键词:域名生成算法;卷积网络;LSTM

中图分类号:TP391 **文献标志码**:A

引用格式:刘洋,赵科军,葛连升,等.一种基于深度学习的快速 DGA 域名分类算法[J]. 山东大学学报(理学版),2019,54(7):106-112.

A fast DGA domain detection algorithm based on deep learning

LIU Yang[1], ZHAO Ke-jun[1,2*], GE Lian-sheng[1], LIU Heng[3]

(1. Informatization Office, Shandong University, Jinan 250100, Shandong, China; 2. School of Computer Science and Technology, Jinan 266237, Shandong, China; 3. Zhongdian Great Wall Internetworking System Application Co., Ltd, Beijing 102209, China)

Abstract: A CNN-LSTM-Concat fast DGA domain classification algorithm based on deep learning is proposed. The multi-layer one-dimensional convolution networks are used to serialize domain name characters. The LSTM network layer is used to enhance the long-distance dependence between characters. By converting the multi-sequence input of LSTM into a single vector input, the training and detection speed can be greatly improved under the premise of ensuring the detection performance. Experiments show that our method has a precision of 98.32% for DGA domain classification using public datasets. At the same time, the detection time is 6.41 times faster than the LSTM method when the accuracy is higher than the epidemic LSTM methods.

Key words: DGA; CNN; LSTM

0 引言

早期的僵尸网络,为了方便受控僵尸主机和命令与控制(C2)服务器之间的相互通信,一般会将 C2 服务器的 IP 地址硬编码到恶意程序中。这种方法存在明显的缺点,一旦恶意程序被反向编译或者通过流量分析发现了 C2 的 IP 地址,控制者将面临失去对整个僵尸网络控制的风险。当前,僵尸网络普遍采用域名生成算法(DGA)来逃避安全人员的检测:首先 C2 服务器与僵尸网络中受控主机通过使用相同的 DGA 算法生成大量伪随机域名,控制者会注册其中部分的域名指向 C2 服务器。受控主机定期遍历查询生成的 DGA 域名或其子集,通过控制者已经注册的域名获得 C2 服务器 IP 地址。

为了检测 DGA 僵尸网络,安全人员早期使用逆向工程[1]尝试从僵尸网络代码获取 DGA 域名,这种方法实现难度大,并且攻击者可以远程更新 DGA 算法。由于使用 DGA 域名的恶意软件需要发起大量 DNS 查询,所以基于 DNS 流量分析成为一种可选的检测手段[2-5]。但这类技术一般需要基于很大的时间窗口,因

收稿日期:2019-05-06; **网络出版时间**:2019-06-13 17:09
网络出版地址:http://kns.cnki.net/kcms/detail/37.1389.N.20190613.1350.047.html
基金项目:十三五国家重点研发计划(2017YFB0803004); 赛尔网络下一代互联网技术创新项目(NGII20150412)
第一作者简介:刘洋(1976—),女,硕士,工程师,研究方向为数据管理和网络安全. E-mail:yang@ sdu.edu.cn
*通信作者简介:赵科军(1981—),男,博士研究生,工程师,研究方向为网络安全和机器学习. E-mail:zhaokejun@ sdu.edu.cn

此很难用于实时监测和预防。DGA 域名为了防止与合法域名冲突加入了随机性,这造成 DGA 域名与合法域名存在统计差异,因此从域名自身的角度进行分类检测成为当前的主流方法[6-9]。这类方法通常使用手工方式收集域名字符的多项特征,包括长度、熵信息、语音辅音比、*n*-gram 信息等,然后构建机器学习分类模型,区别合法域名与 DGA 域名。特征提取是一项困难的工作,需要做大量统计分析,并且对于新出现的 DGA 类型通常需要提取新的特征,这造成该类方法很难发现新的 DGA 域名,并且对新出现的基于字典生成 DGA 域名检测效果较差,分类准确率较低。

深度学习[10]具有模型容量大、无需特征提取等特点,在自然语言处理、图像识别等多种应用场景取得了非常显著的成绩,基于深度学习的 DGA 域名检测成为新的热点方法[11-14]。当前研究人员主要基于循环神经网络(如长短期记忆网络 LSTM[15])对域名进行序列化分类,但由于循环网络需要依赖上一步计算结果,很难有效利用 GPU 硬件进行并行加速,所以训练和检测速度较慢。许多研究已经证明卷积网络在使用一维卷积建模序列数据方面表现良好[16-18]。基于此,本文提出一种 CNN-LSTM-Concat 的快速 DGA 域名检测方法,利用一维卷积实现域名字符的序列化处理,LSTM 网络层用于更好地获得域名字符间的长依赖关系,同时我们将 LSTM 的多序列输入拼接为单序列输入。实验证明本方法在取得良好检测效果的同时,能够将检测速度相比 LSTM 方法提升 6.41 倍。

1　相关工作

DGA 僵尸网络通过域名查询的方式获得 C2 服务器信息,生成的域名通常在短时间内使用,并具有相似的生命周期和查询样式[2],因此研究人员可以通过分析 DNS 查询数据来检测 DGA 僵尸网络。Lee 等[19]检查访问已知恶意域名的客户端的 DNS 查询信息,通过与已知恶意域名的顺序关系和统计共性找到未知的恶意域名。Bilge 等[3]从 DNS 查询流量中提取了分属于时间属性、DNS 应答信息、TTL 信息和域名信息四大类别的 15 个特征,然后基于决策树算法判别僵尸网络的查询行为。类似地,周昌令等[20]从域名查询行为角度出发,在域名的多样性、时间性、增长性和相关性等方面提取 18 种特征数据,基于随机森林算法分类异常域名。Grill 等[21]提出了一种统计方法,使用收集到的 NetFlow/IPFIX 统计数据来检测 DGAs。Kwon 等[4]主要基于 PSD(傅立叶变换)信号处理技术、光谱密度测试技术对用户域名查询行为进行分析,利用查询行为时序关系检测非法域名,同时在处理速度方面得到提高。DGA 僵尸网络在试图联系 C2 服务器的过程中,会产生大量失败域名(NXDomain)回应,Yadav 等[5]首次利用失败域名信息,通过成功域名和失败域名的时间相关性和信息熵特征来检测出被注册的 DGA 域名,进而发现 C2 服务器的地址。Antonakakis 等[22]发现来自同一僵尸网络(具有相同的 DGA 算法)产生类似 NXDomain 流量,首先使用聚类算法根据域名构成和查询这些域名的机器组的相似性进行聚类,然后通过分类算法将生成的簇分配给已知的 DGA 僵尸网络模型。如果聚类得到的某个集群不能分配到已知的模型,那么就可以发现新的僵尸网络类型。NXDomian流量相对整个 DNS 查询流量要小的多,因此基于 NXdomain 特性的检测方法速度更快,模型更轻量化,但是该方法需要一个较大时间窗口的统计信息,无法做到实时检测。

DGA 生成的域名为了避免与合法域名冲突,域名字符带有明显的随机性。Yadav 等[6,23]分析了域名中字母数字 unigrams 和 bigrams 分布的信息熵,将训练集分为两个子集:由 DGA 生成的子集和合法域名子集,计算两个子集的 unigrams 和 bigrams 的分布。分类是分批进行的,每批未知域由共享的二级域和相同 IP 地址的域进行聚类,然后计算每个簇的 unigram 和 bigram 分布,并使用 Kullback-Leibler(KL)距离与两个已知子集进行比较。Phoenix 系统[7]使用了两种基本的语言特征:有意义的字符比和 *n*-gram 正态分值,有意义的字符比率计算域名中包含有意义单词的字符比率。基于域名字符特征和查询 IP 特征与良性集合的 Mahalanobis 距离对未知域名进行分类,一旦域名被分类为 DGA,它们就被发送到 DBSCAN 聚类方法进一步对域名进行细分。Tong 等[24]改进了 Phoenix 算法,利用更多的域名字符特征如信息熵,*n*-grams 出现的频率和域分类的 Mahalanobis 距离等对 DGA 域名进行检测。张维维等[8]使用域名字符蕴含的词素(词根、词缀、拼音及缩写)特征,提出一种轻量级检测算法,能够快速锁定可疑域名,在保证检测正确率的同时,降低了资源消耗率同时提高了检测速度。Truong 等[9]分析大量合法域以及 DGA 域名,发现域名的字符存在明显的统计偏差。基于域名长度和字符信息熵特征,构建 J48 决策树分类算法来区分正常域名和 DGA 生成的域名。上

面的研究需要手工提取这些特征,而这些特征对于不同类别的域名会存在统计信息不显著的特点,因此需要对不同类别的 DGA 域名提取新的特征。而特征提取工程是一项困难工作,并且很难保证能够获得满意的特征。另外基于有限提取的特征,上述 DGA 域名检测方法的准确率也相对较低。

近年来,更多的 DGA 算法正在尝试使用基于字典的伪随机域名,而这些恶意域名更"像"合法域名,如由 Suppobox DGA 生成的"middleapple.net"域名。上面常规的域名检测方法是建立在 DGA 随机生成域名与真实域名语言特征存在显著差异的基础上,因此无法有效的对该类域名进行检测。深度学习[24]在计算视觉、自然语言处理等方向取得了显著的效果,从自然语言处理角度开展基于深度学习的 DGA 域名分类检测成为现在热门的解决方法。Woodbridge 等[11]利用词嵌入技术将域名字符向量化,然后基于长短期记忆网络 LSTM(Long Short-Term Memory)[25]对域名进行分类。LSTM 能够很好地学习到字符间的前后关系,因此该方法在无需手工提取任何特征的情况下,可以准确地分类 DGA 域名。Curtin 等[26]提出了粉碎分数作为域名与英语单词相似性的衡量标准,粉碎分数是域名与英语词典中单词的 n-gram 平均重叠率,n 为 3—5。作者同时使用 WHOIS 查询信息辅助 LSTM 分类,该方法在类字典 DGA 分类中取得更好的效果。合法域名通常包含多个在上下文中有关联的单词,而 DGA 域名由于随机性,缺乏这种关联特性。Koh 等[27]基于以上发现,使用了上下文敏感的嵌入模型 ELMo,该嵌入模型可以区分不同单词上下文并为多义词提供不同表示的词嵌入,将词嵌入层与一个全连接层结合执行域名分类任务。由于 DGA 训练样本中存在类别不平衡问题,Tran 等[28]提出了一种成本敏感 LSTM.MI 算法,将成本项引入到 LSTM 的反向传播机制中,以此提高 LSTM 对不平衡问题的鲁棒性。采用循环神经网络 LSTM 的方法能够很好的发现域名字符间的长距离依赖关系,准确率也大幅超过了传统基于字符统计特征方法。但是由于循环神经网络依赖前一步计算结果,无法充分利用 GPU 实现大规模并行运算,所以训练速度和检测速度较慢。另外 LSTM 方法应用于 DGA 域名容易造成过拟合问题。卷积神经网络(CNN)被越来越多的应用于自然语言处理中[17,29],最近 Huang 等[12]使用 n-gram 将域名字符向量化,利用多层 res-CNN 对 DGA 域名进行分类。卷积网络具有结构相对简单,可以利用 GPU 硬件加速的特点。

2 模型

本文提出基于卷积神经网络与 LSTM 相结合的方式分类检测 DGA 域名,一维卷积可以视为元语法探测器,层次化卷积中后层网络可以获得更长距离的 n 元语法信息。同时在保证准确率的前提下,将 LSTM 的输入由多步循环处理转换为单步处理,大幅提高了训练和检测速度。模型的算法框架如图 1 所示。

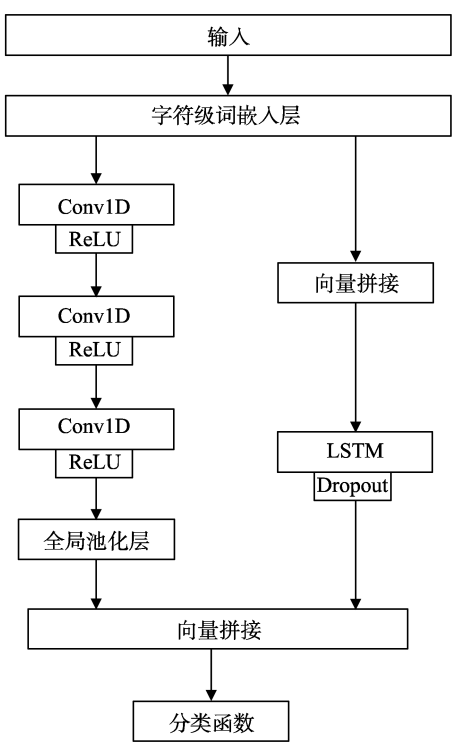

图 1 算法框架

Fig.1 Algorithm Structure

DGA 算法域名主要生成二级域名,所以我们使用二级域名作为模型的输入(如无特别说明,后面域名将特指二级域名)。对于一个域名如"www.google.com","com"是一级域名,"google"是二级域名。首先我们将域名基于字符向量化,字符向量化常用的方法包括词袋(bag-of-words)编码、独热(one-hot)编码、n 元语法(n-gram)等编码方式。n 元语法通过设置多个 n 值得到字符串的向量表示,该方法在早期简单自然语言模型中常用,但是较大的 n 值会导致向量空间过大,从而无法使用较大的 n 值,同时也会造成数据稀疏性问题。此外本文中使用的一维卷积可以看作是 n 元语法提取器,通过多层卷积可以获得更大范围 n 元语法信息。词袋编码使用一个固定长度的向量表示字符串,向量长度为所有出现字符集合的长度,向量每一位代表某一个字符在该字符串中出现的次数。词袋模型由于只记录了字符串中出现的字符,字符间的语义关系和顺序信息没有保留,而这些信息在 DGA 域名检测中是非常重要的。

本研究中,首先统计域名中出现的所有字符生成字典,对于任意给定域名 $A(a_1,a_2,\cdots,a_m)$,转换为长度

为 l 的向量表示,l 为训练数据中最长域名的长度。向量的每一位为域名字符在字典中的索引值,长度小于 l 的域名后面补 0。该编码方式的好处是保留了词袋模型简单高效的特点,同时保存了字符间的前后顺序信息。经过向量化后的域名,经过词嵌入层转换为一个字符向量序列 $\boldsymbol{w}_{1:l}=\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_l$,其中$\boldsymbol{w}_i\in\mathbf{R}^d$,$d$ 为嵌入层向量维度。经过前期超参验证嵌入层向量维度取值对最终结果影响较小,d 在本文中取值 128,可以保留足够的上下文信息。词嵌入层将域名的字符使用密集的字符向量表示,字符向量是通过模型训练过程中得到的,这样我们通过词嵌入可以获得更多隐藏的字符间上下文关系。

本文使用一维卷积实现对域名字符序列化处理。一维卷积使用宽度为的滤波器(或者称为卷积核)在序列向量上平移,每次对 k 个输入序列执行卷积计算:

$$\boldsymbol{x}_i=\oplus(\boldsymbol{w}_{i:i+k-1}),\tag{1}$$

$$c_i=g(\boldsymbol{x}_i\cdot\boldsymbol{u}+b),\tag{2}$$

$$\boldsymbol{x}_i,\boldsymbol{u}\in\mathbf{R}^{k\cdot d},\ c_i,b\in\mathbf{R}。$$

操作符$\oplus(\boldsymbol{w}_{i:i+k-1})$表示拼接向量 $\boldsymbol{w}_i,\cdots,\boldsymbol{w}_{i+k-1}$,$\boldsymbol{u}$ 为滤波器中的权重向量,b 是偏置变量。卷积计算通过滤波器中的权重向量 $\boldsymbol{u}$ 与窗口内经过拼接的向量 $\boldsymbol{x}_i$ 内积,加上偏置结果 b 后由一个非线性函数 g 处理后输出,这里非线性函数 g 采用 ReLU 函数。每层使用 n 个滤波器 $\boldsymbol{u}_1,\cdots,\boldsymbol{u}_n$,组成滤波器矩阵 $\boldsymbol{U}$,得到每一次的输出 $\boldsymbol{c}_i$:

$$\boldsymbol{c}_i=g(\boldsymbol{x}_i\cdot\boldsymbol{U}+\boldsymbol{b}),\tag{3}$$

$$\boldsymbol{x}_i\in\mathbf{R}^{k\cdot d},\ \boldsymbol{c}_i,\boldsymbol{b}\in\mathbf{R}^n,\ \boldsymbol{U}\in\mathbf{R}^{k\cdot d\times n}。$$

本文采用三层卷积网络,各包含 128,256,128 个滤波器,随后紧接一个全局池化层。使用全局池化层替代全连接层减少了参数量,同时也减少了模型的过拟合问题。多层卷积可以扩大视野,从元语法角度分析就是可以获得更大值的语法信息。为了更好获得域名字符间的长依赖关系,我们并行了一个 LSTM 网络层,LSTM 是一种精心设计的循环神经元结构,它将存储的信息和神经元本身结合在一起,通过门控函数可以自动对输入,输出,记忆内容控制。LSTM 很好的解决了普通循环神经元梯度消失问题,能够获取序列中的长依赖信息。由于 LSTM 需要循环使用上一步计算结果,使得多序列 LSTM 的运行时间相比卷积计算时间要长很多。因此我们将输入序列 $\boldsymbol{w}_{1:l}$拼接为单个向量 $\boldsymbol{e}=\oplus(\boldsymbol{w}_{1:l})$,$\boldsymbol{e}\in\mathbf{R}^{l\cdot d}$,通过该操作我们将个序列向量输入转换成了单个向量输入到 LSTM 层,后期实验证明在保证效果的同时使得计算速度得到大幅提高。为了减少过拟合问题,我们在 LSTM 层后加上了 Dropout 层,丢弃率设为 0.5。最后,本研究将卷积层与 LSTM 层的输出拼接,直接输入到分类函数分类,分类函数为 Sigmoid 函数。二元交叉熵作为评价分类结果的损失函数

$$L(\hat{\boldsymbol{y}},\boldsymbol{y})=-\frac{1}{N}\sum_i^N[\hat{y}_i\log\hat{y}_i+(1-y_i)\log(1-\hat{y}_i)]。\tag{4}$$

向量 $\hat{\boldsymbol{y}}$ 为 Sigmoid 函数得到的预测概率,向量 $\boldsymbol{y}$ 为实际目标值,这里 DGA 域名值为 1,合法域名值为 0,使用 Adam[30] 算法最小化损失函数。

3　实验结果与分析

所有实验均使用 Python 语言完成开发,机器学习分类算法使用 Scikit-learn[31] 实现,版本号为 0.19.2;深度学习平台采用 Keras,版本号为 2.2.1,张量库使用 TensorFlow[32],版本号为 1.4.1。实验开发环境为 Ubuntu16.04 x64 操作系统, 64 GB 内存,另外使用一块 NVIDIA 1080Ti 显卡用于深度学习训练、测试加速。

3.1　实验数据

实验数据包括合法域名集和 DGA 域名集,为方便实验重构验证,所有数据均从网络公开数据集获取。Alexa 前 10 万访问域名作为合法数据集,我们选取 bambenek consulting[33] 提供的 DGA 域名 10 万条作为 DGA 域名集,其中包括 Cryptolocker、Locky、Gameover Zeus 等典型的 DGA 域名,也包含 Suppobox、Volatile 等基于字典生成的 DGA 域名。我们将合法域名与 DGA 域名混合打乱,随机抽取 80%数据即 16 万条域名作为训练集,剩下的 20%即 4 万条域名作为测试集。

3.2　试验设计及指标评价

本实验中,DGA 域名为正样本、合法域名为负样本,采用准确率(accuracy)、精确率(precision)、召回率

(recall)、F1 值四个指标评价分类器的分类性能,定义如下:

$$\text{accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}, \tag{5}$$

$$\text{precision}=\frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{6}$$

$$\text{recall}=\frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{7}$$

$$\text{F1}=\frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}}, \tag{8}$$

其中 TP 表示样本为正,预测结果为正;FP 样本为负,预测结果为正;TN 表示样本为负,预测结果为负;FN 表示样本为正,预测结果为负。

对比算法中,Truong 等[9]手工提取域名长度和字符信息熵特征,使用 J48 决策树进行分类,在实验中标识为“J48”;Woodbridge[11]基于 LSTM 算法进行分类,在实验中标识为“LSTM”;对于本文提出的算法,我们标识为“CNN-LSTM-Concat”,为了对比普通 LSTM 与本文采用的输入拼接为单向量 LSTM 的性能,我们将本文算法 LSTM 部分修改为为普通 LSTM,标识为“CNN-LSTM”。此外将本文提出算法中 CNN 部分提取出来作为对比算法,标识为“CNN”。J48 算法采用 Scikit-learn 实现,其余算法均采用 Keras 实现。

3.3 实验结果与对比分析

图 2 是训练过程中各算法在训练集和测试集上准确率与损失函数值变化情况。图中标识以“_testing”结尾的数据表示是在测试集获取的数据,以“_training”结尾表示是在训练集获取的数据。从图中可以看出,基于深度学习的方法在训练集上都能够很好做到拟合。训练收敛速度上,基于 CNN 的算法群在第 20 轮训练前都已经收敛,LSTM 算法在 41 轮收敛。很明显基于 CNN 的算法收敛速度更快,这是因为 CNN 结构相比 LSTM 更简单,对于 DGA 域名分类这种字符分类场景,可以更快速的获得足够特征信息来拟合训练数据,而 LSTM 循环输入方式,更适合长文本依赖关系分析场景。由于基于 CNN 的算法在训练集上拟合速度更快,所以在测试集上的过拟合出现的轮次也相对 LSTM 算法更早些,CNN 算法群在前 10 轮依次出现过拟合问题,LSTM 算法从第 12 轮训练开始出现过拟合问题。

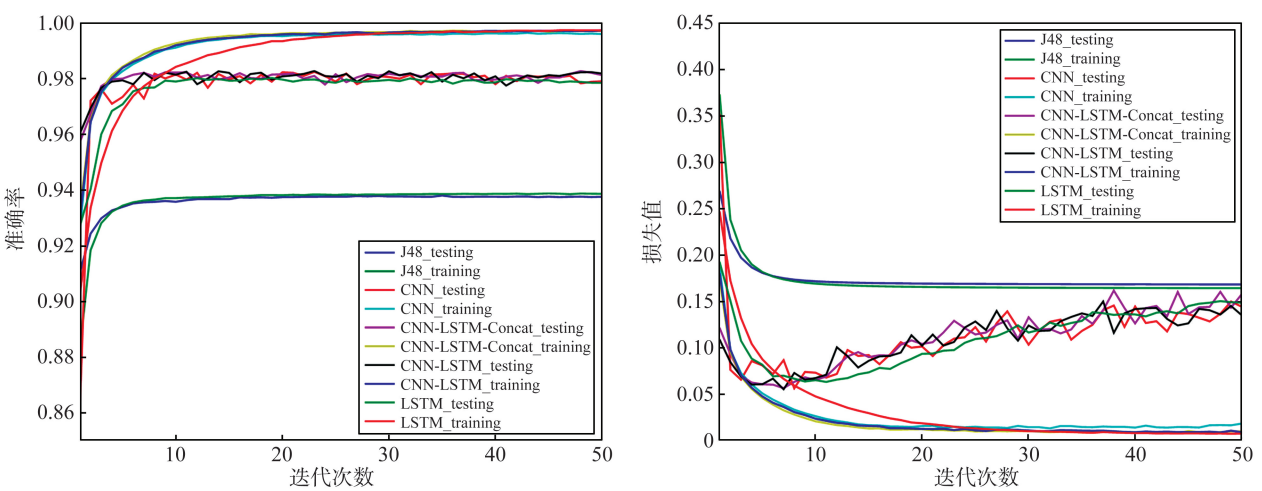

图 2 训练、测试过程中准确率和损失函数值迭代情况

Fig.2 Iteration of accuracy and loss values of training and tesing

表 1 是不同方法在测试集上的分类性能比较。J48 算法的准确率、精确率、召回率和 F1 值分别是 93.81%、95.75%、92.17%和 0.9369,可以明显看出其与基于深度学习分类算法性能上的差距。基于字符特征的传统算法通过提取数个具有明显统计差异的特征进行分类,这对于单个 DGA 域名分类可能是有效的,但是针对多类别 DGA 域名的情况下,几乎无法找到一致具有强分辨力的特征,这也造成该类型的方法难以在实验任务中取得良好的分类效果。基于深度学习的算法各项指标明显得到提升,其中 LSTM 算法和 CNN 算法性能差异较小,准确率、精确率、召回率都达到 98%以上。CNN 与 LSTM 混合模型指标得到进一步提升,本文

提出的 CNN-LSTM-Concat 算法精确率最高,达到 99.4%,回归率可以达到在 99.29%。CNN-LSTM 算法在准确率、召回率和 F1 值取得最佳效果,分别达到 98.32%、99.37%和 0.983 62,CNN-LSTM 算法在这些指标上稍强于 CNN-LSTM-Concat,分别高出 0.01%、0.08%和 0.000 07。这是由于 CNN-LSTM 算法中 LSTM 部分对域名中字符是依次循环处理,相比 CNN-LSTM-Concat 中对域名字符采用单次处理机制能够获得更丰富的特征信息,整体性能前者稍强于后者,但差距非常小。

表 1　几种分类器的性能比较

Table 1　Comparison of classifcatons'performance

	J48	LSTM	CNN	CNN-LSTM-Concat	CNN-LSTM
准确率/%	93.81	98.03	98.12	98.31	**98.32**
精确率/%	95.75	98.55	98.55	**99.41**	99.38
召回率/%	92.17	98.54	98.54	99.29	**99.37**
F1	0.938 92	0.980 27	0.981 66	0.983 19	**0.983 26**

表 2 显示了深度学习算法训练和测试所耗费时间,训练时间为在 16 万训练数据上单次迭代时间的平均值,测试时间为在 4 万测试数据上单次测试平均值。CNN 算法所用时间均为最少,CNN-LSTM-Concat 算法训练时间和测试时间分别为 18.11 秒、4.623 秒,比 CNN 算法耗时略长,说明 LSTM 部分对整体运行速度影响较小。而 LSTM 算法和 CNN-LSTM 算法耗时明显大于 CNN-LSTM-Concat 算法。这是因为 LSLM 和 CNN-LSTM 算法中的 LSTM 部分都是采用循环输入方式,即对域名字符逐个处理,很难实现并行加速。CNN-LSTM-Concat 算法中,我们对输入的域名字符进行拼接,将域名字符当做单向量处理,在保证获取足够的长依赖关系的前提下,大大提高了运行速度。针对分类性能最佳的两种算法 CNN-LSTM-Concat 和 CNN-LSTM,CNN-LSTM-Concat 训练速度比 CNN-LSTM 快 7.28 倍,测试速度快 6.41 倍。CNN-LSTM-Concat 算法在我们普通商用计算机实验环境中平均每秒可处理 8 652 条域名,由于测试流程与现实检测流程是一致的,所以 CNN-LSTM-Concat 可以满足校园网级别网络实时分类检测 DGA 域名的要求。

表 2　几种深度学习分类器运行时间比较

Table 2　Comparison of deep learning classifiers' running time　　s

	LSTM	CNN	CNN-LSTM-Concat	CNN-LSTM
训练	122.780	13.710	18.110	131.790
测试	28.647	3.126	4.623	29.658

4　结论

本文介绍了一种基于深度学习的 DGA 域名检测方法 CNN-LSTM-Concat。我们使用一维卷积网络实现对域名字符序列化处理,通过多层不同核值的卷积核达到更大视野范围。为了更好地获得域名字符间长距离依赖关系,我们并行地加入了一层 LSTM 循环神经网络。同时为了减少 LSTM 神经元计算过程中前后依赖关系对计算时间的影响,我们将输入的字符序列拼接为单个向量,这样 LSTM 的计算过程由多步转换为单步,在保证分类性能的同时,大大减少了对 CNN 并行处理速度的影响。实验表明本文提出的算法在准确率指标上达到 98.31%,各项分类指标都高于 LSTM 算法的情况下,测试时间比 LSTM 算法快 6.41 倍。

参考文献:

[1] STONE-GROSS B, COVA M, GILBERT B, et al. Analysis of a botnet takeover[J]. IEEE Security & Privacy Magazine, 2011, 9(1):64-72.

[2] CHOI H, LEE H, LEE H, et al. Botnet detection by monitoring group activities in DNS traffic[C]// 7th IEEE International Conference on Computer and Information Technology (CIT 2007).[S.l.]:[s.n.], 2007: 715-720.

[3] BILGE L, SEN S, BALZAROTTI D, et al. Exposure: a passive DNS analysis service to detect and report malicious domains [J]. ACM Trans Inf Syst Secur, 2014, 16(4):14:1-14:28.

[4] KWON J, LEE J, LEE H, et al. PsyBoG: a scalable botnet detection method for large-scale DNS traffic[J]. Computer Networks, 2016, 97:48-73.

[5] YADAV S, REDDY A L N. Winning with DNS failures: strategies for faster botnet detection[C]// Security and Privacy in Communication Networks. Berlin: Springer, 2011: 446-459.

[6] YADAV S, REDDY A K K, REDDY A L N, et al. Detecting algorithmically generated malicious domain names[C]// Pro-

ceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. New York: ACM, 2010: 48-61.
[7] SCHIAVONI S, MAGGI F, CAVALLARO L, et al. Phoenix: DGA-based botnet tracking and intelligence[C]// Detection of Intrusions and Malware, and Vulnerability Assessment. Cham: Springer, 2014: 192-211.
[8] 张维维, 龚俭, 刘茜等. 基于词素特征的轻量级域名检测算法[J]. 软件学报, 2016, 27(9):2348-2364.
ZHANG Weiwei, GONG Jian, LIU Qian, et al. Lightweight domain name detection algorithm based on morpheme features [J]. Journal of Software, 2016, 27(9):2348-2364.
[9] TRUONG D-T, CHENG G. Detecting domain-flux botnet based on DNS traffic features in managed network[J]. Security and Communication Networks, 2016, 9(14):2338-2347.
[10] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553):436-444.
[11] WOODBRIDGE J, ANDERSON H S, AHUJA A, et al. Predicting domain generation algorithms with long short-term memory networks[J/OL]. arXiv: 1611.00791 [cs], 2016.
[12] HUANG J, WANG P, ZANG T, et al. Detecting domain generation algorithms with convolutional neural language models[C]// 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). [S.l.]:[s.n.], 2018: 1360-1367.
[13] ZHAUNIAROVICH Y, KHALIL I, YU T, et al. A survey on malicious domains detection through DNS data analysis[J]. ACM Comput Surv, 2018, 51(4):67:1-67:36.
[14] YANG L, LIU G, ZHAI J, et al. A novel detection method for word-based DGA[C]// SUN X, PAN Z, BERTINO E. Cloud Computing and Security. [S.l.]: Springer International Publishing, 2018: 472-483.
[15] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8):1735-1780.
[16] KIM Y. Convolutional neural networks for sentence classification[J/OL]. arXiv: 1408.5882 [cs], 2014.
[17] KARIM F, MAJUMDAR S, DARABI H, et al. LSTM fully convolutional networks for time series classification[J]. IEEE Access, 2018, 6:1662-1669.
[18] KÜHRER M, ROSSOW C, HOLZ T. Paint it black: evaluating the effectiveness of malware blacklists[G]// STAVROU A, BOS H, PORTOKALIDIS G. Research in Attacks, Intrusions and Defenses. Cham: Springer International Publishing, 2014, 8688:1-21.
[19] LEE J, KWON J, SHIN H J, et al. Tracking multiple C&C botnets by analyzing DNS traffic[C]// 2010 6th IEEE Workshop on Secure Network Protocols. [S.l.]:[s.n.], 2010: 67-72.
[20] 周昌令, 陈恺, 公绪晓等. 基于 Passive DNS 的速变域名检测[J]. 北京大学学报(自然科学版), 2016, 52(03):396-402.
ZHOU Changling, CHEN Kai, GONG Xuxiao, et al. Detection of fast-flux domains based on passive DNS analysis[J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2016, 52(03):396-402.
[21] GRILL M, NIKOLAEV I, VALEROS V, et al. Detecting DGA malware using NetFlow[C]// 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). [S.l.]:[s.n.], 2015: 1304-1309.
[22] ANTONAKAKIS M, PERDISCI R, NADJI Y, et al. From throw-away traffic to bots: detecting the rise of DGA-based malware [C]// Proceedings of the 21st USENIX Conference on Security Symposium. Berkeley: USENIX Association, 2012: 24-24.
[23] YADAV S, REDDY A K K, REDDY A L N, et al. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis[J]. IEEE/ACM Transactions on Networking, 2012, 20(5):1663-1677.
[24] TONG V, NGUYEN G. A method for detecting DGA botnet based on semantic and cluster analysis[C]// Proceedings of the Seventh Symposium on Information and Communication Technology-SoICT'16. Ho Chi Minh City, Viet Nam: ACM Press, 2016: 272-277.
[25] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8):1735-1780.
[26] CURTIN R R, GARDNER A B, GRZONKOWSKI S, et al. Detecting DGA domains with recurrent neural networks and side information[J/OL]. [2018-10-04].https://arxiv.org/abs/1810.02023v1
[27] KOH J J, RHODES B. Inline detection of domain generation algorithms with context-sensitive word embeddings[C]// 2018 IEEE International Conference on Big Data (Big Data). [S.l.]:[s.n.], 2018: 2966-2971.
[28] TRAN D, MAC H, TONG V, et al. A LSTM based framework for handling multiclass imbalance in DGA botnet detection [J]. Neurocomputing, 2018, 275:2401-2413.
[29] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification[G]// CORTES C, LAWRENCE N D, LEE D D. Advances in Neural Information Processing Systems 28. [S.l.]:[s.n.], 2015: 649-657.
[30] KINGMA D P, BA J. Adam: a method for stochastic optimization[J/OL]. arXiv:1412.6980 [cs], 2014.
[31] OSINT feeds from bambenek consulting[EB/OL]. https://scikit-learn.org/stable/index.html
[32] Keras[EB/OL]. https://github.com/fchollet/keras
[33] OSINT feeds from bambenek consulting[EB/OL]. [2019-04-20]. http://osint.bambenekconsulting.com/feeds/.

(编辑:李晓红)