计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191000118

基于人工特征与深度特征的 DGA 域名检测算法

胡鹏程[1]　刁力力[2]　叶　桦[1]　仰燕兰[1]
1 东南大学自动化学院　南京 210096
2 趋势科技核心技术部　南京 210012
(pengchenghu@seu.edu.cn)

摘　要　当前,各种各样的恶意软件常使用域名生成算法(Domain Generation Algorithms,DGAs)来生成大量的随机域名,然后尝试与 C&C 服务器建立通信,发动相应的攻击。现有的检测方法基于 DGA 域名的随机性构建人工特征,利用机器学习方法学习分类模式,但该类算法存在人工构建特征费时费力、检测误报率高等问题;或利用 LSTM,GRU 等深度学习技术学习 DGA 域名的序列关系,但该类算法对低随机性的 DGA 域名的检测准确率较低。文中提出了一种域名通用特征的提取方案,建立了包含 41 种 DGA 域名家族的数据集,并设计了基于人工特征与深度特征的检测算法,提高了模型的泛化能力,增加了对 DGA 域名家族的识别种类。实验结果表明,基于人工特征与深度特征的 DGA 域名检测算法取得了比传统深度学习方法更高的准确率和更好的泛化能力。

关键词:域名生成算法;域名检测;长短期记忆网络;特征工程
中图法分类号　TP393.0

DGA Domains Detection Based on Artificial and Depth Features

HU Peng-cheng[1],DIAO Li-li[2],YE Hua[1] and YANG Yan-lan[1]
1 School of Automation,Southeast University,Nanjing 210096,China
2 Core Technology-Research,Trend Micro China Development Center,Nanjing 210012,China

Abstract　Nowadays,various families of malware use domain generation algorithms (DGAs) to generate a large number of pseudo-random domain names to connect to C&C (Command and Control) servers,in order to launch corresponding attacks. There are two existing methods to detect DGA domains. On the one hand,it is a machine learning method based on the randomness of DGA domain name to construct artificial features. This kind of algorithm has the problems of time-consuming and laborious artificial feature engineering and high false alarm rate and so on. On the other hand,LSTM,GRU and other deep learning technologies are used to learn the sequence relationship of DGA domain names. This kind of algorithm has a low detection accuracy for DGA domain names with low randomness. Therefore,this paper proposes a domain name generic feature extraction scheme,establishes a data set containing 41 DGA domain name families,and designs a detection algorithm based on artificial features and depth features that enhances the generalization ability of the model and improves the identification types of DGA domain families. Experimental results show that DGA domain name detection algorithm based on artificial features and depth features has achieved higher accuracy and better generalization ability than traditional deep learning methods.

Keywords　Domain generation algorithms,Domain name detection,Long short-term memory,Feature engineering

1　引言

许多恶意软件包含 DGAs,使得预先防护十分困难。恶意软件会批量生成伪随机域名,然后尝试与 C&C(Command and Control)服务器建立通信。一旦连接成功,恶意软件就会在该服务器上进行更新,上传收集信息并进行其他恶意活动。传统的规则方法是建立 DGA 域名黑名单库,首先判断域名是否在黑名单中,然后决定是否连接该域名,从而抵御后续的攻击。但是,随着动态生成域名的速率和算法种类的增加,这种防御措施变得越来越困难。

Kührer 等对黑名单方法的有效性进行了全面的分析[1],提出了公共黑名单在 DGA 覆盖率方面的严重不足,测试的 DGA 域名只有不到 1.2%包含在任何一个黑名单中,同时恶意软件家族的结果参差不齐,覆盖范围为 0～99.5%。以上研究表明,黑名单是有用的,但是必须补充其他技术,以提供更加充分的保护。

除了使用黑名单过滤方法,还有构建 DGA 分类器的方法,该分类器可以在网络中找出 DNS 请求并查找 DGAs。当检测到 DGAs 时,分类器会通知其他自动化工具或网络管理员进一步调查 DGA 的起源。传统的 DGA 检测工作可以分

到稿日期:2019-10-18　返修日期:2020-03-11　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
通信作者:叶桦(zhineng@seu.edu.cn)

为两类:回溯检测和实时检测。

回溯检测对大集合域名进行批量预测,这种方法不能用于实时检测和防御的系统[2-4]。在这些系统中,使用聚类技术将域名集合分组,并生成每个分组的统计属性。分类器通过在训练过程中生成的模板和统计测试,对潜在的 DGAs 组进行分类。此外,这些技术还结合了上下文信息,如 HTTP 包头、跨网络的 NxDomains 和被动 DNS,来进一步提高性能,但是不能满足许多需要实时检测和防御的实际安全应用程序的需求。此外,对于许多安全应用程序来说,使用上下文信息是不现实的,例如终端检测和响应(Endpoint Detection and Response, EDR)系统运行在终端和主机上,对设备的处理性能、网络稳定性和内存容量有严格的要求。每个终端从网络收集上下文信息需要太多的带宽和时间开销,这在实际应用中难以实现。

实时检测仅使用域名(没有其他上下文信息),将域名划分成正常域名和 DGA 域名。与回溯检测相比,实时检测是一个相当困难的问题,而回溯检测的性能对于实际部署应用来说往往太低。传统的实时检测方法是提取域名特征,利用机器学习区分正常域名和 DGA 域名。常用的特征有熵、域名长度[5]、元音与辅音的比例等,这些特征被输入机器学习模型中,如随机森林、SVM 等。使用人工特征主要有两个缺点:首先,人工设计的特征容易被绕过;其次,设计人工特征是一个非常耗时的过程,需要随着域名生成算法的更新而更新。如果黑客通过攻击一组特征来派生出一个新的 DGA 家族,安全人员则需要花费相当多的时间来创建新的特征。

随着深度学习在图像、语音和自然语言等方面取得重大突破,深度学习技术应用得越来越广泛。Woodbridge 等利用长短期记忆网络(Long Short-Term Memory, LSTM),Lison 等使用门控循环单元(Gated Recurrent Unit, GRU),来检测恶意域名[6-7];CHEN 等[8]将注意力机制引入双向循环神经网络中构建检测模型。这些利用循环神经网络的方法对随机性高的 DGA 域名具有较高的准确率,但对随机性较低的 DGA 域名的识别率较低,因此不能做到对绝大部分 DGA 家族都有很好的检测准确率。

本文提出了一种基于人工特征和深度特征的 DGA 域名实时检测方法,即通过融合人工提取的通用特征与 LSTM 自动提取的深度特征构建模型,降低了识别的误报率。相比传统的特征工程加机器学习方法,这项技术缩短了人工设计特征的耗时,提高了识别的准确率。相比单一的深度学习方法(LSTM, GRU, ATT-GRU 等),本文提出的方法加入了人工经验特征,提高了模型的泛化能力,增加了 DGA 域名家族的识别种类。

2 域名生成算法

本文对 41 种不同类型的恶意软件中域名生成算法生成的域名进行分类能力的评估。恶意软件家族包括勒索软件(如 Cryptolocker[9] 和 Cryptowall[10])、银行木马(如 Hesperbot[11])和一般信息窃取策略(如 Ramnit[12])。

DGA 技术的复杂性各不相同,有简单的统一生成的域名,也有试图在真实域中模拟分布的域名。例如,Ramnit 使用在种子[12]上计算的一系列除、乘和取模等创建域名,而 Suppobox 通过连接两个随机字符串(通常取自英语)[13]创建域。

(1)基于算术的 DGA

根据时间或者随机种子,初始化出一系列可以根据 ASCII 码直接表示成域名的值,或者使用这些值作为偏移量,指向 DGA 硬编码的字符表中的一个字符。目前,网络上大部分的 DGA 域名都是由这种算法生成的。

(2)基于哈希的 DGA

使用十六进制表示的哈希值生成 DGA 域名,通常有 SHA256 和 MD5 两种哈希值。

(3)基于单词表的 DGA

从一个或者多个单词表中随机选择单词,并将其拼接成一个域名。

(4)基于置换的 DGA

对正常的域名进行置换操作,生成多个新的域名。

3 基于人工与深度特征的域名检测算法

本文在分析多种域名生成算法实现机制的基础上,设计了一种域名通用特征的提取方案,包括域名的基本特征、高级特征以及网络安全领域特征,并在构建的数据集上分析人工特征的有效性。数据集包含 41 种常见的域名生成算法生成的 1 199 949 条真实的恶意域名和 1 000 000 条合法网站域名,详情请见 4.1 节中的数据集构建。

3.1 域名特征提取

3.1.1 基本特征

随机性是域名特征提取的基本特征。

香农熵公式提供了一种基于符号编码字符串的平均最小比特数的方法。

$$H(x)=-\sum_{i=0}^{N-1} p_i \log_2 p_i$$

其中,p_i 表示给定字符出现的概率。

香农熵可以衡量域名的随机性。如图 1 所示,正常域名香农熵整体分布在 DGA 域名的左边,意味着正常域名的随机性较低,而 DGA 域名的随机性相对较高。

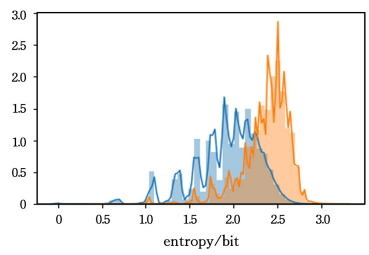


图 1 正常域名与 DGA 域名的香农熵分布

Fig. 1 Normal and DGA domains Shannon entropy distribution

仅依靠香农熵可以区分 google.com 与 vnaisdhfv.cn,因为前者包含重复字符,随机性相对较低,但正常域名和 DGA 域名之间并非是绝对的差距,还需要其他更加高级的特征补充。

3.1.2 高级特征

(1)是否能读

正常的网站为了实现更多的浏览量,往往将域名设计成容易朗读且容易记住的形式。

1)元音字母的比重

元音是指在发音过程中由气流通过口腔而不受阻碍发出的音。英文中共有 5 个元音，分别是 a e i o u。此类形式的常见网站有 google.com，baidu.com 等，

为了方便用户阅读，正常域名都会将元音字母包含进域名中，而 DGA 域名具有随机生成的特性，并且元音字母在英文字符中的占比较低，因此 DGA 域名中元音字符的占比通常较低。正常域名与 DGA 域名中元音字母的比重分布如图 2 所示。

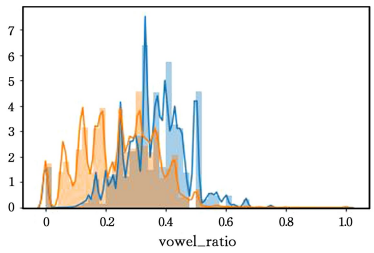


图 2 正常域名与 DGA 域名中元音字母比重分布

Fig. 2 Normal and DGA domains vowel ratio distribution

2)Gibberish

判断一个字符是否能被真实发音，还可以通过 gibberish detection 来量化[14]，如在谷歌网站未被大家知晓前，“google”并不是一个单词，但是人们却能很顺利地读出来。

Gibberish 的原理是马尔可夫链，首先在英文语料库上进行训练，记录各字符出现的频率。例如考虑文本“Tom study every day”，gibberish 计算 To om m[space] [space]s… 这些双字节对出现的频率，当处理完所有训练文本后，将计数归一化，那么每个字符在给定初始值之后就有 27 个后续字符(26 英文字母＋空格)的概率分布；然后给定一个字符串，gibberish 通过将字符串中相邻字符对的概率相乘来衡量该字符串出现的概率。正常域名与 DGA 域名的 gibberish 分布如图 3 所示。

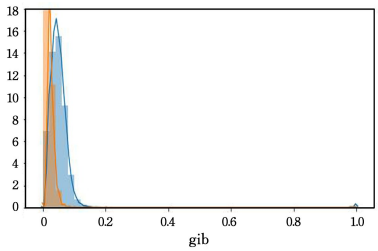


图 3 正常域名与 DGA 域名的 gibberish 分布

Fig. 3 Normal and DGA domains gibberish distribution

(2)连续与分散

通过研究进一步发现，DGA 域名的随机性也表现在字母和数字的分布上。一般地，随机生成的 DGA 域名都不会出现连续的数字或者相同的字母；而正常域名为了吸引眼球，会以纯数字或者独特的数字组合加入域名中，如 360.com，hao123.com 等。同时，因为英文字母辅音字符远多于元音字符，DGA 域名更可能连续出现辅音字符；而合法的域名为了便于发音，会交替使用辅音和元音字符。

正常域名与 DGA 域名数字字符占比、重复字符占比、连续字符占比、连续辅音字符占比的分布如图 4－图 7 所示。

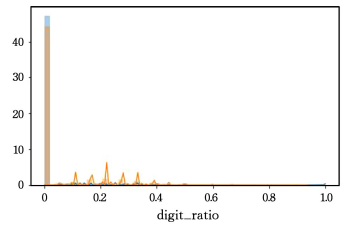


图 4 正常域名与 DGA 域名数字字符占比分布

Fig. 4 Normal and DGA domains digit ratio distribution

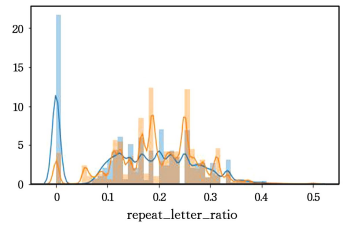


图 5 正常域名与 DGA 域名重复字符占比分布

Fig. 5 Normal and DGA domains repeat letter ratio distribution

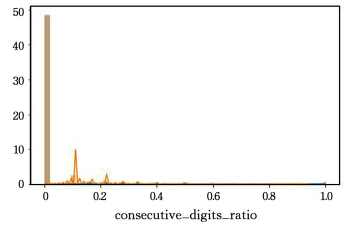


图 6 正常域名与 DGA 域名连续字符占比分布

Fig. 6 Normal and DGA domains consecutive digits ratio distribution

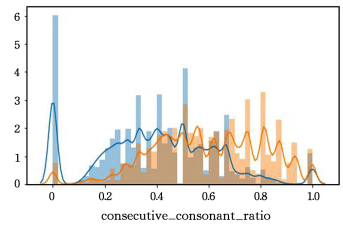


图 7 正常域名与 DGA 域名连续辅音字符占比分布

Fig. 7 Normal and DGA domains consecutive consonant ratio distribution

(3)n-gram

在深度学习被广泛应用之前，研究机器学习的学者处理自然语言时常采用 uni-gram(单字)、bi-gram(双字)和 tri-gram(三字)等手段，n-gram 往往能提供重要的信息。

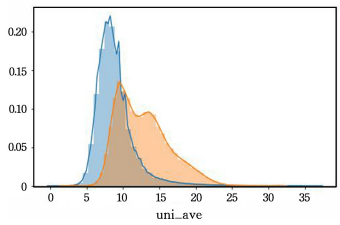
首先，将一段正常英文文本拆成单字、双字和三字；然后，统计各种组合出现的频次，根据频次对出现的组合进行排名，形成常用字符组合排名表[15]，^ 表示字符串开始，$ 表示字符串结束。表 1 所列为在当前数据集中排名前五的单字、双字和三字。

表 1 单字、双字和三字的 Top-5

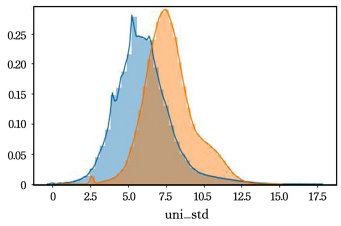
Table 1 Uni-gram，bi-gram and tri-gram Top-5

Rank	Uni-gram	Bi-gram	Tri-gram
1	e	s $	ing
2	a	er	er $
3	o	in	es $
4	i	an	ent
5	r	ar	log

正常域名与 DGA 域名单字节排名的均值和方差、双字节排名的均值和方差、三字节排名的均值和方差分别如图 8—图 10 所示。



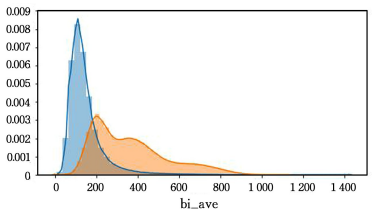
(a)



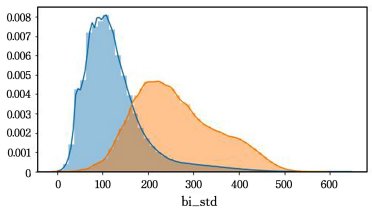
(b)

图 8 正常域名与 DGA 域名单字排名的均值和方差

Fig. 8 Mean and variance of normal and DGA domains uni-gram rank



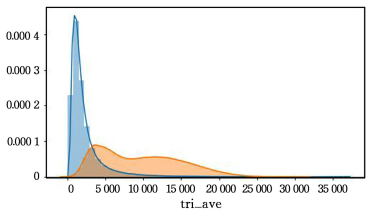
(a)



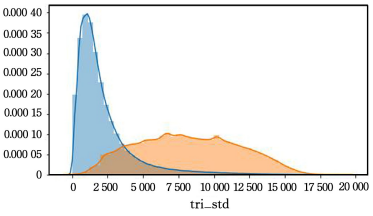
(b)

图 9 正常域名与 DGA 域名双字排名的均值和方差

Fig. 9 Mean and variance of normal and DGA domains bi-gram rank



(a)



(b)

图 10 正常域名与 DGA 域名三字排名的均值和方差

Fig. 10 Mean and variance of normal and DGA domains tri-gram rank

(4)其他

隐马尔可夫模型是关于时序的概率模型，描述由一个隐藏的马尔可夫链随机生成不可观测的状态随机序列，再由不可观测的状态随机序列生成可观测的随机序列的过程[16]。正常域名与 DGA 域名的 log_HMM 分布如图 11 所示。

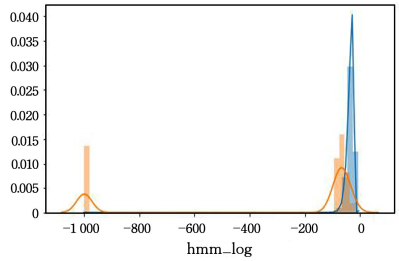


图 11 正常域名与 DGA 域名的 log_HMM 分布

Fig. 11 Normal and DGA domains log_HMM distribution

域名由字符组合而成，可以看作一个序列，因此可以使用隐马尔可夫模型计算域名出现的概率。具体的做法是：首先，取所有的反例样本(合法的域名)，计算双字间的转移概率矩阵；然后，当给定一个新的域名时，根据转移矩阵计算域名的隐含马尔可夫概率。由于连乘小于 1 的数会导致结果极小，在工程上一般采用 log 化方法进行处理。

$$p(S)=p(w_1w_2\cdots w_n)=\prod_{i=0}^{n-1}w_i$$

其中，w_i 表示 A_i 到 A_{i+1} 的转移概率。

3.1.3 领域特征

(1)ccTLD

在通常情况下，.com 等顶级域名的申请是十分昂贵和困难的，因此 DGAs 往往会使用. biz. info. ru 等容易申请的 ccTLD，但这种情况并非 100%准确。现实中的 ccTLD 成千上万，并且 ccTLD 之间不存在大小关系，如果采取 One-Hot 编码，那么特征维度将极大，造成维度灾难。在工程中的做法是，先将 ccTLD One-Hot 编码，然后对此稀疏矩阵进行 SVD 降维操作，一般降到 15～30 维，实验中使用的是 30 维。

(2)域名长度

随着时间的发展，DGAs 生成的域名变得越来越长。

3.2 深度特征提取

在各种自然语言处理任务中，循环神经网络(Recursive Neural Networks，RNNs)常常被用来捕捉序列中有意义的短时关系[17-18]。RNNs 的主要优点在于它将上下文(状态)信息引入输入输出的映射中，即单个 RNN 的输出是由输入和以前的 RNN 输出决定的。但自循环连接引入的连乘操作链使得传统 RNN 的输出会使给定输入呈指数衰减，从而导致梯度消失问题，这使学习长期依赖关系变得困难。

长短期记忆网络是一种常见的循环神经网络，使用时间反向传播训练，解决了一般 RNNs 的梯度消失问题[19-20]。LSTM 可以用来创建更大、更深的循环神经网络，以解决复杂的序列问题。

图 12 为 LSTM 门控循环单元示意图，下文称其为存储单元。存储单元包含自有状态和输出门，并根据输入的序列来操作每一个存储单元，存储单元内的每个门通过 sigmoid 激活函数来控制是否被触发。

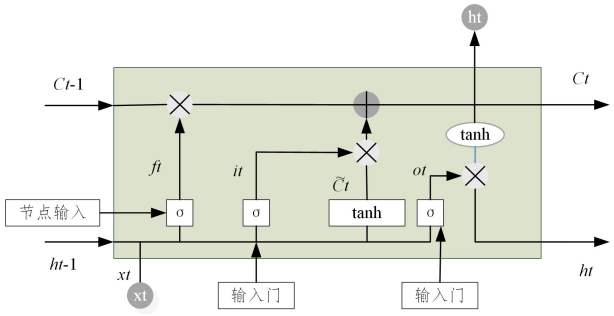


图 12　LSTM 神经单元

Fig. 12　LSTM cell unit

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f)$$

$$i_t=\sigma(W_i\cdot[h_{t-1},x_t]+b_i)$$

$$\widetilde{C}_t=\tanh(W_C\cdot[h_{t-1},x_t]+b_C)$$

$$C_t=f_t*C_{t-1}+i_t*\widetilde{C}_t$$

LSTM 长期依赖关系的学习方式非常适合处理文本、语音和语言方面的任务。在本文实现的方法中，使用它们来学习域名序列的深度特征。提取深度特征部分的深入网络如图 13 所示。

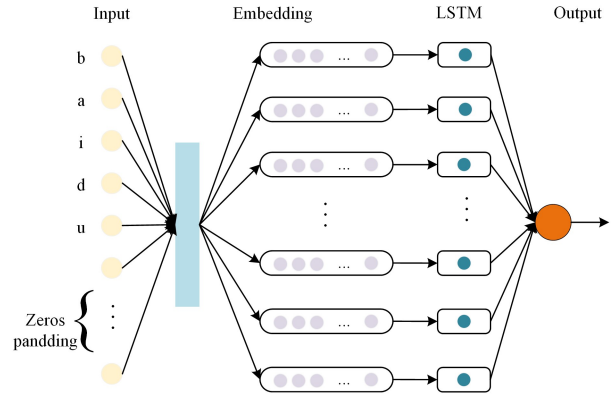


图 13　LSTM 提取深度特征的网络结构

Fig. 13　LSTM network of extracting depth feature

首先对域名进行统一长度的编码，不足的部分用 0 补充。深度的第一层是 Embedding layer，其作用是将编码后的输入映射到统一的空间。第二层网络是包含 128 个 LSTM 存储单元的网络层，用于提取域名序列的高级特征。至此，本文设计的特征提取部分全部完成。

3.3　融合模型框架

在实际机器学习应用和近年的热门数据科学竞赛中，集成模型方法在绝大多数情况下都会带来或多或少的帮助。融合后的结果取决于融合模型的数量、单个模型的分类能力和模型间的差异性等要素。本文提出的基于人工特征与深度特征的 DGA 域名检测算法，本质上是将人工特征和深度特征同时加入网络，以补充深度网络无法提取的额外信息，提高了深度模型的泛化能力。

图 14 给出了模型的总体架构。融合前的左半部分的输入是编码后的域名原始序列，用于提取深度特征信息；右半部分的输入是人工提取的 45 个通用特征，用于补充深度网络无法提取的额外信息。

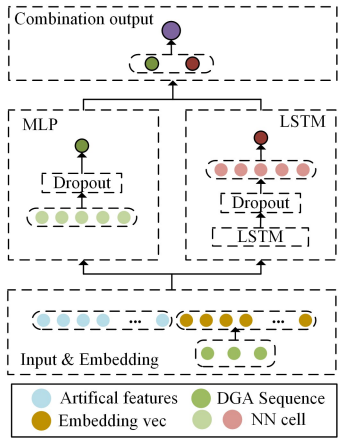


图 14　模型总体结构

Fig. 14　Model overall architecture

4　实验与结果分析

4.1　数据集构建

实验抓取的正样本[1)]包括来自 41 个 DGA 家族的 1 199 949 条真实的 DGA 域名，负样本取自 Alexa 网站根据浏览量排名的前 1 000 000 条合法网站域名。正负样本的比例为 6∶5，分布基本均衡，并且正样本基本囊括了常见的域名生成算法家族。表 2 列出了数据集中占比较大的域名家族；表 3 列出了实验中使用的数据集的具体信息，包括采样数据集（sampling dataset）和全量数据集（all dataset）。

表 2　数据集分布

Table 2　Data set distribution

Rank	Family	Quantity	Ratio/%
1	legit	1 000 000	45.455 6
2	banjori	452 341	20.561 4
3	emotet	330 562	15.025 9
4	rovnix	179 935	8.179 1
5	tinba	72 490	3.295 1
6	pykspa_v1	44 278	2.012 7
7	simda	23 731	1.078 7
8	ramnit	18 935	0.860 7
9	gameover	11 997	0.545 3

表 3　数据集划分

Table 3　Data set partitioning

Name	Train	Validation	Test
sampling	80 000	20 000	20 000
all	1 407 967	351 992	439 990

4.2　评估指标的设计

对于二分类任务，通常使用下述几种评估指标。

$$Precision=\frac{TP}{TP+FP}$$

$$Recall=\frac{TP}{TP+FN}$$

$$F_1=2\cdot\frac{Precision\cdot Recall}{Precision+Recall}$$

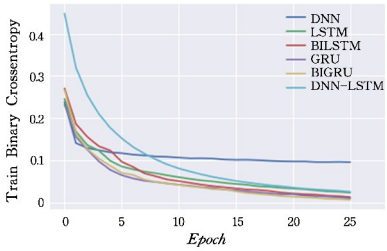
4.3　实验结果与分析

4.3.1　模型对比

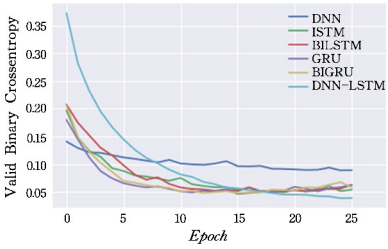
本文对比了传统的机器学习方法（如 LR 和 SVM）、集成

1) http://data.netlab.360.com/feeds/dga/dga.txt

学习(如 LightGBM)和深度学习(如 DNN, LSTM, ATT_LSTM, GRU, ATT_GRU 等)。上述模型使用 4.1 节中采样后的训练集进行训练,其中 LR, SVM, LightGBM, DNN 使用人工特征,深度网络使用域名序列,如图 13 所示。训练过程如图 15 所示,评估指标结果如表 4 所列。



(a)训练集损失曲线



(b)验证集损失曲线

图 15 训练集和验证集损失值的变化趋势

Fig. 15 Trends of train's and validation's loss

表 4 不同方法的准确率、召回率和 F_1 值

Table 4 Precision, recall and F_1 score of different algorithms

Name	*Precision*/%	*Recall*/%	F_1
Logistic	96.1861	95.1864	0.9568
SVM	97.2327	96.4172	0.9682
LightGBM	98.9348	96.5266	0.9772
DNN	97.7019	97.2833	0.9749
LSTM	98.7921	98.4228	0.9861
GRU	98.8732	98.3955	0.9863
ATT_LSTM	98.8538	98.2861	0.9857
ATT_GRU	98.6572	98.4593	0.9856
Arti&Deep	**99.3932**	**99.5596**	**0.9897**

基于人工特征与深度特征的检测算法(DNN_LSTM)迭代到第 15 轮附近时,在验证集上的损失开始低于其他的算法,而此时在训练集上的损失仅低于基于人工特征的前馈神经网络,说明人工特征附加的额外信息提高了深度模型的泛化性能。

分析表 4,可以得出如下结论:1)传统的 LR, SVM 都取得了不错的分类性能,说明本文人工提取的通用特征具有不错的效果;2)集成学习(LightGBM)取得了与深度学习相近的成绩,这也是在最近热门的数据科学竞赛中,集成学习往往成为参赛选手的第一选择原因;3)在样本充足的情况下(sampling 数据集有 100000 个样本),深度学习比传统机器学习方法表现得要好;4)基于人工特征和深度特征的检测算法相比单一的深度学习算法,增加了人工提取的额外特征信息,提高了模型的泛化能力。

4.3.2 错误分析

通过 4.3.1 节的模型分析,本文提出的基于人工特征与深度特征的 DGA 域名检测算法具有误报率低、泛化性高等优点。为了进一步提高模型的识别精度,使用所有收集的数据(即 large 数据集)来训练模型,并使用五折交叉验证工程技巧对训练出的模型进行进一步的集成,提高了模型的稳定性和准确率。

表 5 列出了 5 折交叉验证过程中产生的 5 个单模型及集成模型在测试集上的精确度、召回率和 F_1 值。其中,集成模型采用的算术平均方案为:

$$y_{\text{ensemble}}=\frac{1}{n}\sum_{i=1}^{n}y_i$$

表 5 全量数据五折交叉验证的准确率、召回率和 F_1 值

Table 5 Precision, recall and F_1 of five folds on all data

Name	*Precision*/%	*Recall*/%	F_1
fold_1	99.6792	9.4425	0.9956
fold_2	99.6240	99.4691	0.9955
fold_3	99.6422	99.4346	0.9954
fold_4	99.6071	99.5100	0.9956
fold_5	99.5794	99.5396	0.9956
ensemble	**99.7032**	**99.5371**	**0.9962**

集成模型在精确率、召回率和 F_1 值等指标上都有明显的提升。表 6 列出了五折集成模型在测试上的合法域名与各个 DGA 家族分类的准确率和支持度。

表 6 全量数据五折集成模型在测试集上的准确率和支持度

Table 6 Accuracy and support of dataset with fivefolds model on all data

Family	*Accuracy*/%	*Support*	Family	*Accuracy*/%	*Support*	Family	*Accuracy*/%	*Support*
legit	99.644	200000	symmi	100.000	851	fobber_v2	98.305	59
banjori	100.000	90468	shifu	91.732	508	feodo	100.000	53
emotet	99.977	66112	**suppobox**	**79.285**	**420**	tempedreve	80.000	40
rovnix	99.994	35987	qadars	98.500	400	pykspa_v2_real	97.435	39
tinba	99.662	14498	locky	90.517	232	padcrypt	94.117	34
pykspa_v1	99.345	8856	dyre	100.000	200	**matsnu**	**0**	**26**
simda	99.936	4746	cryptolocker	99.000	200	bamital	100.000	21
ramnit	96.144	3787	chinad	100.000	200	proslikefan	90.000	20
gameover	100.000	2399	pykspa_v2_fake	85.000	160	**vidro**	**65.000**	**20**
ranbyus	99.908	2181	dircrypt	98.026	152	gspy	100.000	19
virut	**78.332**	**1943**	**vawtrak**	**71.140**	**149**	**mydoom**	**66.666**	**9**
murofet	99.941	1712	**conficker**	**77.319**	**97**	omexo	100.000	8
necurs	94.070	1636	nymaim	91.764	85	tinynuke	100.000	6
shiotob	99.435	1594	fobber_v1	100.000	59	tofsee	100.000	4

用本文设计的算法对 41 种 DGAs 家族进行识别评估,识别率在 80%以上的有 35 种,识别率在 95%以上的有 27 种,识别率在 99%以上的有 22 种。因此,本文设计的算法对绝大部分的 DGAs 具有较高的识别准确度。但是,也存在部分

识别率不太理想的 DGAs 家族，如 virut，suppobox，vawtrak 和 matsnu 等。从识别结果来看，matsnu 的识别率为 0，这是由于 matsnu 的生成机制是从两个正常的单词表中依次选择单词拼接成最终域名，不存在字母级别的随机性，因此本文提及的算法对此类 DGA 域名的识别效果不理想。此外，mydoom 和 suppobox 也是基于单词表的 DGA 算法，而 vawtrak 是基于哈希的算法。

从收集的样本分布情况来看，上述几种 DGA 域名较少，因此可以从以下两方面进行改进。一方面，收集更多的识别效果不好的 DGA 族样本；另一方面，可以对不确定的样本进行二次识别。

结束语 本文提出基于人工特征与深度特征的 DGA 域名检测算法，将人类经验量化的特征输送到模型中，与深度特征一同训练，弥补了深度网络无法提取额外信息的缺陷。

从实验结果来看，与现有算法相比，基于人工特征与深度特征的算法模型增强了模型的泛化能力，进而降低了识别的误报率，提高了召回率，模型的整体性能优于现有算法。

参 考 文 献

[1] KÜHRER M，ROSSOW C，HOLZ T. Paint it black：Evaluating the effectiveness of malware blacklists[C]// International Workshop on Recent Advances in Intrusion Detection. Cham：Springer，2014：1-21.

[2] ANTONAKAKIS M，PERDISCI R，NADJI Y，et al. From Throw-Away Traffic to Bots：Detecting the Rise of DGA-Based Malware[C]// 21th USENIX Security Symposium. 2012.

[3] YADAV S，REDDY A K K，REDDY A L N，et al. Detecting Algorithmically Generated Malicious Domain Names[C]// Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement 2010. Melbourne，Australia，ACM，2010.

[4] KRISHNAN，TAYLOR T，MONROSE F，et al. Crossing thethreshold：Detecting network malfeasance via sequential hypothesis testing[C]// 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE Computer Society，2013.

[5] MOWBRAY M，HAGEN J. Finding Domain-Generation Algorithms by Looking at Length Distribution[C]// IEEE International Symposium on Software Reliability Engineering Workshops. IEEE，2014.

[6] WOODBRIDGE J，ANDERSON H S，AHUJA A，et al. Predicting domain generation algorithms with long short-term memory networks[J]. arXiv：1611. 00791，2016.

[7] LISON P，MAVROEIDIS V. Automatic detection of malware-generated domains with recurrent neural models[J]. arXiv：1709. 07102，2017.

[8] CHEN L H，CHEN H，FANG Y Q. Detecting Domain Generation Algorithm Based on Attention Mechanism. [J]. Journal of east China University of Science and Technology (Natural Science Edition)，2019(3).

[9] LIAO K，ZHAO Z，DOUPEA，et al. Behind closed doors：measurement and analysis of CryptoLocker ransoms in Bitcoin[C]// Electronic Crime Research. IEEE，2016.

[10] SULKOSWKI A J. Cyber-Extortion：Duties and Liabilities Related to the Elephant in the Server Room[J/OL]. SSRN Electronic Journal. https：//ssrn. com/abstract=955962.

[11] ATZENI A，DIAZ F，LOPEZ F，et al. The Rise of Android Banking Trojans[J]. IEEE Potentials，2020，39(3)：13-18.

[12] ALBANESIUSC. Ramnit computer worm compromises 45K facebook logins[J/OL]. http：//www. pcmag. com/article2/0.

[13] PLOHMANN D，YAKDAN K，KLATT M. A comprehsivemeasurement study of domain generatingmalware[C]// 25th USENIX Security Symposium. Austin：Usenix，2016：263-278.

[14] Gibberish-Detector[OL]. https：//github. com/rrenaud/Gibberish-Detector.

[15] DGA feature mining[OL]. https：//www. cnblogs. com/bonelee/p/7640055. html.

[16] LI H. Statistical learning methods [M]. Beijing：Tsinghua University Press，2012.

[17] ROBINSON A J. An application of recurrent neural nets to phone probability estimation[J]. IEEE Trans. on Neural Networks，1994，5(2)：298-305.

[18] BENGIO Y，BOULANGER-LEWANDOWSKI N，PASCANU R. Advances in optimizing recurrent networks[C]// 2013 IEEE International Conference on Acoustics，Speech and Signal Processing. IEEE，2013.

[19] GRAVES A. Long Short-Term Memory[M]// Supervised Sequence Labelling with Recurrent Neural Networks. 2012.

[20] GERS F A，SCHRAUDOLPHN N，SCHMIDHUBER. Learning Precise Timing with LSTM Recurrent Networks[J]. Journal of Machine Learning Research，2003，3(1)：115-143.

HU Peng-cheng，born in 1996，postgraduate. His main research interests include pattern recognition and intelligent system，data mining，deep learning，network security，etc.

YE Hua，born in 1961，Ph.D. His main research interests include intelligent control，pattern recognition，computer application，intelligent building，intelligent robot，fault diagnosis，etc.