计算机工程与应用
Computer Engineering and Applications
ISSN 1002-8331,CN 11-2127/TP

《计算机工程与应用》网络首发论文

题目：　改进 Relief-C5.0 的恶意域名检测算法
作者：　马栋林，张澍寰，赵宏
网络首发日期：　2021-04-16
引用格式：　马栋林，张澍寰，赵宏．改进 Relief-C5.0 的恶意域名检测算法．计算机工程与应用. https://kns.cnki.net/kcms/detail/11.2127.TP.20210416.1010.006.html

网络首发时间：2021-04-16 13:00:36
网络首发地址：https://kns.cnki.net/kcms/detail/11.2127.TP.20210416.1010.006.html

改进 Relief-C5.0 的恶意域名检测算法

马栋林，张澍寰，赵宏

兰州理工大学 计算机与通信学院，兰州 730050

摘　要：针对目前恶意域名检测算法中分类模型计算复杂度较大、实时性不强以及准确率不高等问题，提出了 Rf-C5(Relief-C5.0)恶意域名检测算法模型。首先，提取待测域名的全局 URL 特征，根据提取的特征按照改进的 Relief 算法进行权重计算，并依据权重值进行优先级排序；然后选取权重值排名前 20 的关键特征作为 C5.0 分类器的输入端，进行合法域名与恶意域名的分类。实验结果表明，在大样本数据集下，Rf-C5 模型与当前主流恶意域名检测算法相比，在提高平均检测速率的基础上，检测准确率提高了 1.58%-4.91%。

关键词：恶意域名；URL 特征；改进的 Relief 算法；C5.0 分类器

文献标志码：A　中图分类号:TP391　doi：10.3778/j.issn.1002-8331.2012-0475

Malicious Domain Names Detection by Improved Relief-C5.0

MA Donglin, ZHANG Shuhuan, ZHAO Hong

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Abstract: Aiming at the problems of the high computational complexity, low real-time performance, and low accuracy of classification models in the current malicious domain name detection algorithms, a malicious domain name detection algorithm by Rf-C5(Relief-C5.0) is proposed. Firstly, the global URL features of the domain names to be tested are extracted. Then, the improved Relief algorithm was used to calculate the weight of the extracted features, and the features are prioritized according to the weight values. Finally, the key features of the top 20 weighted values are selected as the input of C5.0 classifier to classify legitimate domain names and malicious domain names. Experimental results show that under the large sample data set, compared with the current mainstream malicious domain name detection algorithms, the detection accuracy of Rf-C5 model increases by 1.58%-4.91% on the basis of increasing the average detection rate.

Key words: malicious domain name; URL features; improved Relief algorithm; C5.0 classifier

域名系统(Domain Name System，DNS)作为互联网的一项基础服务，提供域名和 IP 地址之间的相互转换。此外，DNS 还作为信任的凭据，为邮件服务器和证书时域控制权提供验证服务。由于 DNS 应用广泛，且自身缺乏安全检测机制，因此成为恶意域名主要的攻击对象。

恶意域名攻击常伴随邮件发送、短信和网页点击等内容中，通过使用一些迷惑性的文字和图片来引诱用户点击，也可能以某种形式存在于软件代码中，伴随软件的运行对某一服务器发起大规模访问，导致服

基金项目：国家自然科学基金(61262016)；赛尔网络下一代互联网技术创新项目(NGII20160311, NGII20160112)。

作者简介：马栋林(1971－),男,副教授,研究方向为深度学习、网络空间安全;张澍寰(1994－),男,硕士研究生,E-mail:599239440@qq.com;赵宏(1971－)，男，博导，教授，CCF 会员。

务器宕机，影响合法域名的正常访问。利用恶意域名检测技术，及时发现网络中出现的恶意域名并进行拦截，可以有效防范恶意域名的攻击。

根据国家互联网应急中心(National Internet Emergency Center, CNCERT)2020 年第 51 期《网络安全信息与动态周报》显示[1]，该周内因恶意域名攻击导致境内感染网络病毒的主机数量达 75.8 万个，境内被篡改网站数量达 4327 个，被植入后门的网站数量达 839 个，境内网站的仿冒页面数量达 4374 个，可见由恶意域名引起的网络安全形势不容乐观。

在众多恶意域名攻击中，僵尸网络攻击[2]所产生的负面作用尤为巨大。目前，僵尸网络的控制者大部分使用 DGA(Domain Generation Algorithm)算法来生成域名，从而逃避黑名单的检测。国内外的研究者如 Mao 等[3]提出了一种针对 DGA 的恶意域名检测算法，应用机器学习方法提取特征集，构建 DGA 检测模型。Can 等[4]将域名数据模糊化为 Neutorsophic 集，来减少良性域名的误检。Sivaguru 等[5]提取域名的边界信息特征，与域名进行内联，构造深度学习框架，并使用随机森林算法进行分类。殷聪贤等[6]利用随机森林算法构建了基于 DNS 行为特征的恶意域名检测模型。该算法因使用的特征过多，导致时间开销较大，此外，该模型对训练数据要求极高，若存在干扰数据，随机森林模型会出现过拟合现象，导致检测准确率不稳定。Zhao 等[7]提出了一种基于词法分析和特征量化的恶意域名检测算法。该算法首先根据待测域名和黑名单上域名之间的编辑距离，将待测域名划分为明确恶意或潜在恶意；然后利用 N-gram 计算潜在恶意域名的信誉值，根据信誉值判断潜在恶意域名的恶意性，通过在公开数据上验证了该方法的有效性。

此外，域名变换技术 Fast-Flux 和 Domain-Flux 也常用来隐藏复杂代理网络背后的恶意服务器，使得恶意域名的状态处于不断变化中。如 Truong 等[8]提出了一种基于被动 DNS 流量跟踪分析的恶意域名检测方法，通过提取十个 URL 关键特征，并利用机器学习算法来建立分类器。崔甲等[9]结合黑/白域名过滤器、DNS 记录解析器以及基于特征分类的检测引擎等三种域名检测技术，构建了新型恶意域名检测框架，具有较好的完备性。

在域名生成和变换算法的基础上，Fu 等[10]提出了一种隐身域名生成算法 SDGA(Stealthy Domain Generation Algorithm)，与传统的基于字符的 DGA 恶意域名检测算法相比，该算法的隐蔽性更强，更难实时捕获访问日志记录。Yang 等[11]利用 SDGA 域名的特征层特征，提出了一种异构深度神经网络框架。采用改进的多尺度卷积核并行 CNN 结构，从域名中提取多尺度局部特征，并加入基于自注意力机制的双向 LSTM 网络架构，提取带有注意力机制的双向全局特征。杨路辉等[12]针对 SDGA 生成的域名难以检测问题，在现有卷积神经网络模型的基础上，增加了提取更深层字符级特征的卷积分支，同时提取恶意域名的浅层和深层字符级特征并融合，提高对复杂样本的检测准确率。

以上恶意域名检测方法各有优势，相比而言，使用机器学习方法检测时，若提取的特征过多，则时间开销较大，若提取的特征较少，则不具备代表性，准确率不高；基于深度神经网络的检测方法准确率较高，但耗时较长，实时性不强；通过建立黑名单过滤器的检测方法耗时较短，准确率也较高，但容易被攻击者所规避，普适性不强。

综上，基于目前恶意域名检测算法中存在的实时性不强、准确性不高等问题，本文提出一种基于 Rf-C5 的恶意域名检测算法。首先，通过使用改进的 Relief 特征选择算法解决了字符特征数量过多造成计算开销大、运行时间长的问题；其次，通过 C5.0 决策树分类器，降低分类的复杂度，在保证检测准确率的基础上降低检测的时间开销。

1 算法设计

图 1 给出了本文算法的检测流程。首先，收集与整理国内外合法域名与恶意域名公开数据集，并进行预处理；然后，整合当前已有特征，并在此基础上增加了域名的全局 URL 构词特征种类，将其作为特征选择层的输入；其次，在特征选择层中使用改进的 Relief 算法计算全局特征的权重，并输出关键特征集，作为分类层的输入；最后，在分类层中使用 C5.0 决策树进行二分类，实现合法域名与恶意域名的检测。

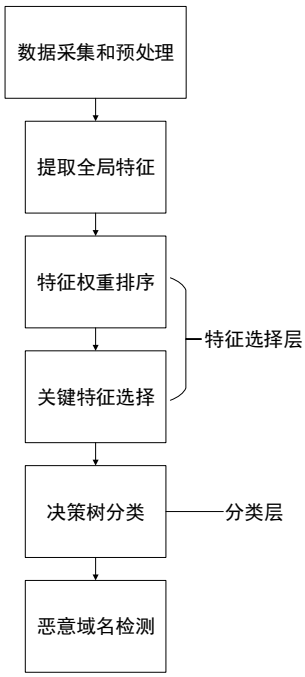

图 1 算法流程

Fig.1 Algorithm flow

1.1 数据采集和预处理

在 Alexa、Malware Domain List 等知名网站上收集与整理合法域名与恶意域名，构造合法域名与恶意域名数据集。此外，为降低噪声数据对检测准确率和实时性的影响，将 http、www 等字符串进行剔除。

1.2 提取全局特征

本文特征提取在文献[13]-[18]等的研究基础上，丰富了特征类别。具体分为字符特征、域名特征、访问特征、TTL(Time To Live)特征、解析特征和 IP 地址集特征六个大类。如表 1 所示。

表 1 全局特征说明与编号

Table 1 Description and numbering of global features

特征类别	特征说明	编号
字符特征	特殊字符/字符串	Q1
	字符随机性	Q2
	字母个数	Q3
	数字个数	Q4
	字母与数字比例	Q5
	元音字母比例	Q6
	分隔符个数	Q7
	唯一字符数	Q8
	唯一字符比例	Q9
	连续字母个数最大值	Q10
	连续数字个数最大值	Q11
	分隔符内字母个数最大值	Q12
	同形异义字符个数最大值	Q13
	最长连续非元音字符串长度比例	Q14
	最长有意义字符串长度比例	Q15
域名特征	域名长度	Q16
	二级域名长度	Q17
	子域名个数	Q18
	域名包含 Level 层数	Q19
	顶级域名随机性	Q20
	是否包含知名域名	Q21
访问特征	平均每日访问时长	Q22
	平均每日访问次数	Q23
	平均每日相似度	Q24
	平均每日变化度	Q25
TTL 特征	TTL 值所处范围	Q26
	TTL 值变化次数	Q27
	TTL 值极差	Q28
	A 记录对应的 TTL 平均值	Q29
	NS 记录对应的 TTL 平均值	Q30
	不同 TTL 值的个数	Q31
解析特征	应答标识位	Q32
	一定时间内域名被解析的总次数	Q33
	应答报文对应不同的 IP 地址个数	Q34
	应答报文 IP 对应的不同国家个数	Q35
	应答报文中应答段/权威段的 RR 个数	Q36
IP 地址集特征	IP 地址集的大小	Q37
	共享同一 IP 地址的域名集的大小	Q38
	请求域名的源 IP 地址集的大小	Q39
	IP 地址集的分散程度	Q40

整合了 15 个字符特征(Q1-Q15)。由于 DNS 的主要功能是为用户提供可读且易于记忆的名称，字符特性如随机性、字母个数和数字个数等与恶意行为紧密关联。域名字符的随机性通过字符的熵来计算，如式(1)所示：

$$H(\mathrm{d}) = -\sum \lg(P(X_i)) \times P(X_i) \quad (1)$$

其中，d 为待测域名；X_i 为 d 中的某一个字符；$P(X_i)$ 为该字符出现的概率。

整合了 6 个域名特征(Q16-Q21)。一般正常域名的长度、随机性、子域名个数等都较为规范，顶级域名也较为常见，比如.cn 和.com 等。而恶意域名较为混乱和随意。

整合了 4 个时间特征(Q22-Q25)。通过分析域名在时间序列上的查询点，可以发现恶意行为的特征。设置每日为一个观测窗口，统计域名在一日之内的状态变化规律。恶意域名一般不会持续保持活跃，查询次数变化范围较大。

整合了 6 个 TTL 特征(Q26-Q31)。TTL 值被用来设定域名响应记录的最长缓存时间。恶意网络会产生频繁的 TTL 变化，其管理者通过设置不同的 TTL 值为僵尸节点分配资源，表现出更加分散的特点。

整合了 5 个解析特征(Q32-Q36)。攻击者所使用的 IP 地址随机性比较高,会将恶意域名解析到不同国家、不同地区的主机上，IP 地址也会在多个不同的域名间共享来躲避封堵。

整合了 4 个 IP 地址集特征(Q37-Q40)。一般请求恶意域名的源 IP 地址集较小，而共享同一 IP 的恶意域名集较大。通过计算 IP 地址的 16bit 前缀的熵来表示 IP 地址集的分散程度，如式(2)所示：

$$IP\_entorpy = -\sum_x p(x) * \log_2 p(x) \quad (2)$$

其中，$p(x) = count(x) / |I|$，I 表示 IP 地址集，$p(x)$ 为 IP 地址的 16bit 前缀 x 在 I 中所占的比例,熵越大，IP 地址越分散。

1.3 特征选择层

本文总结与分析了大量的 URL 构词特征，但实际该类特征中只有一部分是对分类有效的特征，如果使用全部的特征训练，可能会导致以下问题：

(1)特征数量过多导致特征向量维度过高、容易出现模型过拟合；

(2)特征数量过多导致模型训练过程缓慢，影响检测实时性。

因此,本文采用改进的 Relief 算法来对全局特征进行权重排序，根据排序结果选择最佳分类特征，删除冗余特征对于检测结果的影响。同时，选择主流特征选择算法 Filter 相关系数法和 Wrapper 递归特征消除法进行对比实验，验证 Relief 算法的有效性。

1.3.1 Filter 相关系数法

Filter 按照特征与标签的相关性进行评分,并根据动态阈值判别法来选择所需的特征。本文借助机器学习算法中的 sk-learn 模块，使用 feature_selection 库的 SelectKBest 类构建 Filter 相关系数模型，代码如下所示：

SelectKBest(lambda X, Y: array(map(lambda x:pearsonr(x, Y), X.T)).T, k=20). fit_transform(iris.data, iris.target)

该模型的输入为 40 维特征矩阵和标签值,输出为包含了特征和皮尔逊相关系数（P 值）的数组。取 10 次实验的平均 P 值作为特征的排名依据，由于 20 位以后的特征 P 值较低，对检测结果影响较小，本文设定参数 k=20,P 值排名最高的 20 个特征如表 2 所示。

表 2 Filter 关键特征组合

Table 2 Filter key features combination

排名	特征编号	P 值	排名	特征编号	P 值
1	Q5	0.85	11	Q2	0.64
2	Q7	0.84	12	Q3	0.62
3	Q1	0.79	13	Q34	0.59
4	Q13	0.77	14	Q16	0.58
5	Q4	0.74	15	Q10	0.53
6	Q24	0.74	16	Q39	0.48
7	Q12	0.73	17	Q8	0.35
8	Q17	0.70	18	Q15	0.31
9	Q6	0.66	19	Q30	0.28
10	Q27	0.65	20	Q40	0.28

1.3.2 Wrapper 递归特征消除法

递归消除特征法通过使用一个基模型来进行多轮训练，每轮训练后，消除若干权值系数的特征；然后，再次基于新的特征集进行下一轮训练。本文通过借助机器学习算法中的 sk-learn 模块,使用 feature_selection 库的 RFE 类构建 Wrapper 递归特征消除法模型，代码如下所示：

RFE(estimator=LogisticRegression(), n_features_to_select=20).fit_transform(iris.data, iris.target)

该模型的输入为 40 维特征矩阵和标签值,输出为选择的关键特征集。基模型采用 Logistic Regression 逻辑回归函数，为使对比实验结果更加准确可靠，本文设定参数 n_features_to_select=20，将所选择的特征个数与 Filter 关键特征组合保持一致，取 10 次实验的平均值作为检测结果。输出如表 3 所示。

表 3 Wrapper 关键特征组合

Table 3 Wrapper key features combination

排名	特征编号	排名	特征编号
1	Q5	11	Q15
2	Q13	12	Q17
3	Q4	13	Q31
4	Q16	14	Q22
5	Q23	15	Q27
6	Q40	16	Q12

7	Q9	17	Q33
8	Q39	18	Q6
9	Q7	19	Q3
10	Q2	20	Q11

1.3.3 改进的 Relief 算法

Relief 算法根据各特征与标签的相关性计算权重，移除权重小于设定阈值的特征。由于其简洁的算法和优秀的特征选择能力被广泛应用，伪代码如下：

```
Relief (E, m, T)
  Separate E into E+= {positive instances}and E−= {negative instances};
  S= (0, 0, …,0);
 For i=1 to m;
   Pick at random an instance H∈E;
   Pick at random one of the positive instances closest to H, X+∈E+;
   Pick at random one of the negative instances closest to H, X−∈E−;
   if (X is a positive instance);
     then Near-hit= X+; Near-miss= X−;
    else Near-hit= X−; Near-miss= X+;
   update-weight (S, H, Near-hit, Near-miss);
Relevance=(1/m) S;
 For i=1 to N;
   if (relevance_i≥t);
     then f_i is a relevant feature;
     else f_i is an irrelevant feature;
update-weight (S, H, Near-hit, Near-miss);
 For i=1 to N;
   S_i= S_i – diff(h_i, near-hit_i)^2+diff(h_i,near-miss_i)^2;
End.
```

其中参数为数据集 E，抽样次数 m，特征权重阈值 T，特征权重向量 S，样本点 H，正样本集合中距离 H 最近的样本 X^+，负样本集和中距离 H 最近的样本 X^-，特征总个数 N =40。

由于 Relief 算法的运行时间由抽样次数 m 和输入的特征个数 N 决定，计算某一个样本点的最近邻样本需要全部的训练空间，所以存储率较高且运算时间较长。为了解决此问题，本文在 Relief 算法的基础上进行改进，将原始训练集划分为若干个小训练集，分别计算每个小训练集内的特征权重，减小训练空间大小，降低硬件存储率，最后将计算出权重的特征合并，输出关键特征集 T。

根据提取的 40 维全局特征的天然属性，将其分为 6 个小训练集，作为改进的 Relief 算法的输入，流程如图 2 所示。

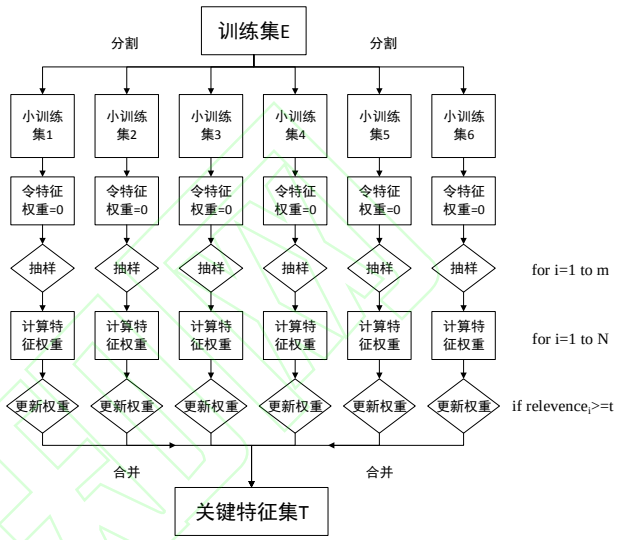

图 2 改进的 Relief 算法

Fig.2 Improved Relief algorithm

由图 2 可以看出，改进的 Relief 算法与原算法的区别在于将数据集分割之后同时并行处理，极大地缩短了运算时间。

取 10 次实验的平均权重值作为检测结果，为了使输出更加直观便于比较，将排名最高的特征权重值归化为 1，其他特征权重值等比例转化。由于后 20 位特征权重值较低，与结果的关联性不强，同样取权重值排名前 20 位的特征作为关键特征，如表 4 所示。

表 4 改进的 Relief 关键特征组合

Table 4 Improved Relief key features combination

排名	特征编号	权重比例	排名	特征编号	权重比例
1	Q5	1	11	Q36	0.61
2	Q1	0.98	12	Q9	0.57
3	Q13	0.92	13	Q34	0.53
4	Q7	0.87	14	Q16	0.44
5	Q22	0.85	15	Q27	0.40
6	Q18	0.81	16	Q4	0.38
7	Q9	0.80	17	Q20	0.34
8	Q17	0.75	18	Q11	0.27
9	Q12	0.72	19	Q2	0.22
10	Q24	0.69	20	Q6	0.09

1.4 分类层

将特征选择层中输出的关键特征组合作为分类层的输入。本文模型在分类层中使用 C5.0 决策树作为分类器，C5.0 以信息熵的下降速度作为确定最佳分支变量和分割阈值的依据。设 S 是一个样本集合，目标变量 C 有 k 个分类，$freq(C_i,S)$ 表示 S 中属于 C_i 类的样本数，$|S|$ 表示样本集合 S 的样本数。则集合 S 的信息熵定义为式(3)：

$$Info(S)=-\sum_{i=1}^{k}((freq(C_i,S)/|S|)\times\log_2(freq(C_i,S)/|S|)) \tag{3}$$

如果某属性变量 T，有 n 个分类，则属性变量 T 引入后的条件熵定义为式(4)：

$$Info(T)=-\sum_{i=1}^{n}((|T_i|/|T|\times Info(T_i)) \tag{4}$$

属性变量 T 带来的信息增益定义为式(5)：

$$Gain(T)=Info(S)-Info(T) \tag{5}$$

随着决策树的生长，越深层处的节点所体现的数据特征就越个性化，会出现过拟合现象，所以需要修剪决策树，采用 Post-Pruning 法从叶节点向上逐层剪枝。一般决策树会使用测试数据进行检验，但 C5.0 分类器使用了统计的置信区间的估计方法，直接在训练数据中估计误差。在执行效率和内存使用方面进行了改进，采用 Boosting 方式提高模型准确率，计算速度较快，占用的内存资源较少。

2 实验与分析

2.1 数据集

实验的数据集包括合法域名和恶意域名两部分。从 Alexa 网站排名中选取前 60000 条域名作为合法域名集，从 DGA Domain List、Malware Domain List 和 360 等知名恶意域名网站上收集并整理 60000 条恶意域名作为恶意域名集，保证了恶意域名种类的完整性和全面性。将合法域名集与恶意域名集进行合并，共 120000 条，选取其中 80000 条作为模型的训练数据，40000 条作为测试数据。

2.2 实验环境

实验环境如表 5 所示。

表 5 实验环境

Table 5 Experimental environment

开发环境	参数
处理器	i9-10850K(全核 4.70GHz)
GPU	RTX3080 10GB
内存	16GB
操作系统	Windows 10 2004(64 位)
IDE	Pycharm、RStudio
开发语言	Python3.8、R-4.0.3

2.3 模型评价标准

使用召回率(Recall)、准确率(Accuracy Rate, AR)、漏报率(False Negative Rate, FNR)、误报率(False Positive Rate, FPR)、精确率(Precision Rate, PR)和 AUC(Area Under Curve)[19]来评估本文提出的 Rf-C5 模型在恶意域名检测时的性能。评价指标均基于实验结果的混淆矩阵计算。

2.4 改进的 Relief 算法效果评估

分别使用 Filter 相关系数法、Wrapper 递归特征消除法和改进的 Relief 算法对提取的 40 维全局 URL 特征（Q1-Q40）进行选择，得到不同的三组特征组合，如表 2-表 4 所示。将其分别作为支持向量机（Support Vector Machine, SVM）、K 最近邻算法（K-Nearest Neighbor,KNN）、随机森林（Random Forest, RF）、C4.5 和 C5.0 五个分类器的输入，交叉验证模型检测效果。图 3-图 5 分别是三种特征选择算法的关键特征组合在不同分类器上的检测结果：

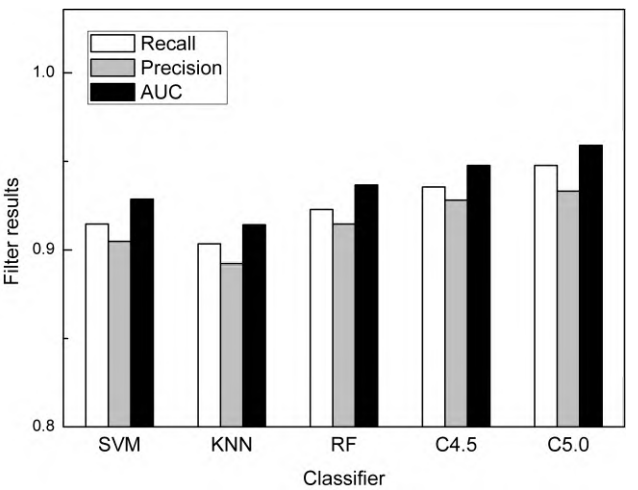

图 3 Filter 特征组合检测结果

Fig.3 Filter feature combination detection results

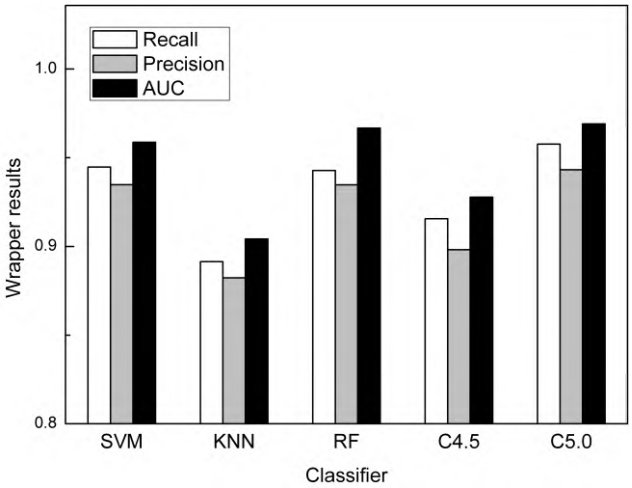

图 4 Wrapper 特征组合检测结果

Fig.4 Wrapper feature combination detection results

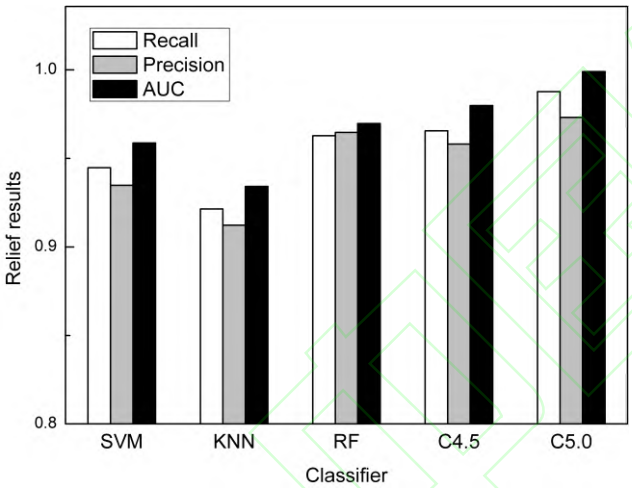

图 5 改进的 Relief 特征组合检测结果

Fig.5 Improved Relief feature combination detection results

由图 3-图 5 可以明显地看出，经过改进的 Relief 算法选择出的特征组合，对恶意域名的检测效果在召回率（Recall）、精确率（Precision）和 AUC 值三个方面都强于 Filter 相关系数法和 Wrapper 递归特征消除法。即改进的 Relief 算法对关键特征的选择更准确，能够更好地提取出与标签关联度更高的特征。

2.5 C5.0 分类器效果评估

将改进的 Relief 算法与 SVM、KNN、RF、朴素贝叶斯(Naive Bayesian, NB)、线性回归(Linear Regression, LiR)、逻辑回归(Logistic Regression, LoR)以及 C5.0 共七种分类器分别结合，构建 Rf-SVM、Rf-KNN、Rf-RF、Rf-NB、Rf-LiR、Rf-LoR 和 Rf-C5 模型，测试不同分类器与改进的 Relief 算法结合之后在域名分类上的效果。

各模型的输入端是提取的 40 维全局 URL 特征，首先通过改进的 Relief 特征选择层计算，输出关键特征集，然后作为分类层的输入，使用分类器进行二元分类。

以 AR(%)、PR(%)、FNR(%)、FPR(%)四项指标深入具体对比七种模型的检测性能，如表 6 所示：

表 6 七种模型的检测性能对比

Table 6 Comparison of detection performance of seven models

算法模型	AR/%	PR/%	FNR/%	FPR/%
Rf-C5	98.79	97.33	1.01	2.19
Rf-NB	94.89	93.77	3.28	2.90
Rf-SVM	93.82	96.74	4.28	3.44
Rf-LiR	94.20	94.12	3.13	3.45
Rf-LoR	97.45	97.01	1.84	2.73
Rf-RF	96.72	93.39	3.72	5.16
Rf-KNN	93.91	95.07	2.41	4.22

由表 6 可以看出，使用 C5.0 分类器与改进的 Relief 算法相结合，在各项指标下均可以达到良好的效果。其中，在准确率方面，C5.0 相比 SVM、KNN、NB 和 LiR 优势明显；在精确率方面，C5.0 相比 NB、LiR 和 RF 优势明显；在漏报率方面，C5.0 相比 NB、SVM、LiR 和 RF 优势明显；在误报率方面，C5.0 相比 RF 和 KNN 优势明显。

2.6 Rf-C5 模型综合效果评估

为了验证 Rf-C5 恶意域名检测模型的综合性能，在相同的实验环境下分别构造目前国内外主流的各类恶意域名检测模型。分别为文献[20]基于语义表示和深度学习的恶意域名检测模型、文献[21]基于 N-grams 和随机森林的恶意域名检测模型、文献[22]基于优化支持向量机的恶意域名检测模型、文献[23]基于卷积神经网络 CNN 的恶意域名检测模型。使用相同的数据集，与本文恶意域名检测模型进行性能比较，具体结果如表 5 所示。

表 7 五种算法性能比较

Table 7 Performance comparison of five algorithms

检测算法	运算时间/s	准确率/%
文献[20]	24.48	94.89
文献[21]	36.89	94.62
文献[22]	42.24	97.21
文献[23]	23.74	93.88
本文算法	9.21	98.79

由表 7 可以看出，本文提出的恶意域名检测算法模型 Rf-C5 相较于主流神经网络和机器学习的检测方法，在运算时间和准确率方面都有显著的提升。

在运算时间方面，由于改进的 Relief 算法将原始数据集划分为若干个小训练集，减小了训练集的计算空间，降低了存储率，提升了运算速度；同时，C5.0 分类器使用统计的置信区间的估计方法，直接在训练数据中估计误差，占用的内存资源较少，与随机森林、支持向量机等分类器相比，极大地减少了运算时间。

在准确率方面，由于 Rf-C5 模型在对域名分类之前先对特征进行了排序选择，删除了无用特征，降低了过拟合；再者，C5.0 决策树分类器采用信息增益率来确定最佳分组变量和最佳分割点，通过 Boosting 方式提高模型准确率，相较于其他分类算法效果更好。从结果来看，准确率略高于文献[22]，优于文献[20]、文献[21]和文献[23]，分别提高了 3.9%、4.17%、1.58%和 4.91%。具有更高的准确性和更好的实时性。

3 结束语

针对目前恶意域名检测算法分类模型计算复杂度较大等问题，构造了一种 Rf-C5 恶意域名检测算法模型。通过对全局特征进行选择，删除了冗余信息；通过与传统的各分类器模型对比，证明了 C5.0 分类器在检测准确率上的优势；进一步在相同的实验环境内，通过与各类主流恶意域名检测模型进行对比，证明了本文 Rf-C5 模型的优良综合性能。在未来的工作里，计划加入多标签分类，可以将良性域名和恶意域名根据内容或功能进一步细分，为用户提供更多有用的信息，提高网络安全。

参考文献：

[1] 网络安全信息与动态周报[EB/OL].(2020-12-23)[2020-12-23]. https://www.cert.org.cn/publish/main/44/2020/20201223142310431885870/20201223142310431885870_.html
Weekly report on Network Security Information and Dynamics[EB/OL].[2020-12-23]. https://www.cert.org.cn/publish/main/44/2020/20201223142310431885870/20201223142310431885870_.html

[2] LIU Z H, ZHANG Y D, CHEN Y Z, et al. Detection of algorithmically generated domain names using the recurrent convolutional neural network with spatial pyramid pooling[J]. Entropy,2020,22(9).

[3] MAO J, ZHANG J M, TANG Z, et al. DNS anti-attack machine learning model for DGA domain name detection[J]. Physical communication,2020,40.

[4] CAN N V, TU D N, TUAN T A, et al. A new method to classify malicious domain name using Neutrosophic sets in DGA botnet detection[J]. Journal of intelligent & Fuzzy systems,2020,38(4): 4223-4236.

[5] SIVAGURU R,PECK J,OLUMOFIN F, et al. Inline detection of DGA domains using side information[J]. Ieeeaccess,2020,8:141910-141922.

[6] 殷聪贤. 基于大数据分析的恶意域名检测技术研究与实现[D].北京：北京邮电大学,2018.
YIN C X. Research and implementation of malicious domains detection technology based on big data analysis[D]. Beijing: Beijing University of posts and telecommunications, 2018.

[7] ZHAO H, CHANG Z B, WANG W J, et al. Malicious domain names detection algorithm based on lexical analysis and feature quantification[J]. Ieee access, 2019, 7:128990-128999.

[8] TRUONG DT, TRAN DT, HUYNH B. Detecting malicious Fast-Flux domains using feature-based classification techniques[J]. Journal of internet technology, 2020, 21(4):1061-1072.

[9] 崔甲,施蕾,李娟,等.一种高效的恶意域名检测框架[J].北京理工大学学报,2019,39(01):64-67.
CUI J, SHI L, LI J, et al. An effective malicious domain detection framework[J]. Transactions of Beijing institute of technology,2019,39(01):64-67.

[10] FU Y, YU L, HAMBOLU O, et al. Stealthy Domain Generation Algorithms (DGAs)[J]. IEEE transactions on information forensics & Security, 2017, 12(6):1430-1443.

[11] YANG L H, LIU G J, DAI Y W, et al. Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework[J]. Ieee access,2020,8: 82876-82889.

[12] 杨路辉,刘光杰,翟江涛, 等.一种改进的卷积神经网络恶意域名检测算法[J].西安电子科技大学学报，2020, 47(01):37-43.
YANG L H, LIU G J, ZHAI J T, et al. Improved algorithm for detection of the malicious domain name based on the convolutional neural network[J]. Journal of xidian university,2020,47(01):37-43.

[13] YAN G H, LI Q, GUO D, et al. Discovering suspicious APT behaviors by analyzing DNS activities[J]. Sensors, 2020, 20(3).

[14] 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.
CHANG Z B. Research on staged malicious domain names detection algorithm based on domain names words formation features[D]. Lanzhou: Lanzhou university of technology, 2020.

[15] ZHANG P P, LIU T W, ZHANG Y. Domain watcher: detecting malicious domains based on local and global textual features[C]//KOUMOUTSAKOS P, LEES M, KRZHIZHANOVSKAYA V, et al. International Conference on Computational Science (ICCS), Zurich, Switzerland,2017,108:2408-2412.

[16] YANG F P, SHENG W T, LONG Y. A Joint approach to detect malicious URL based on attention mechanism[J]. International journal of computational intelligence and applications,2019,18(3), Art. 1950021.

[17] YU B, PAN J, GRAY D, et al. Weakly supervised deep learning for the detection of domain generation algorithms[J]. Ieee access,2019,7(9):51542-51556.

[18] SREYASEE D, ASHIT T, EHAB A. Prioritized active

learning for malicious URL detection using weighted text-based features[C]//ZHENG X D, ZHANG H, XING C X, et al. 15th IEEE International Conference on Intelligence and Security Informatics - Security and Big Data (ISI), Beijing, PEOPLES R CHINA,2017. IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA,2017:107-112.

[19] 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(01):227-231.
ZHAO H, CHANG Z B, WANG L. Fast malicious domain name detection algorithm based on lexical features[J]. Journal of computer applications, 2019, 39(01): 227-231.

[20] 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 & Security,2019,85:77-88.

[21] SELVI J, RODRIGUZE R J, SORIA-OLIVAS E. Detection of algorithmically generated malicious domain names using masked N-grams[J]. Expert systems with applications,2019,124: 156-163.

[22] HUANG J Y, ZHANG G D, SHEN Y J. DGA domain name detection based on SVM under grey wolf optimization algorithm[C]. // WENZHENG L, BABU MSP.10th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, PEOPLES R CHINA, 2019. IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2019: 245-248.

[23] ZHOU S F, LIN L F, YUAN J K, et al. CNN-based DGA detection with high coverage[C]. // ZHENG X, ABBASI A, CHAU M, et al. 17th IEEE Annual International Conference on Intelligence and Security Informatics (ISI), Shenzhen, PEOPLES R CHINA, 2019. IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2019: 62-67.