基于机器学习的恶意 URL 检测

周孟莹
兰州大学，信息科学与工程学院，信息安全
甘肃，兰州 730000

摘要：随着互联网的普及使用，越来越新奇的恶意 URL 攻击出现在网络上，采用黑名单机制的方法已经不能提供完全的保护。对于这些 URL，基于机器学习分类算法的检测技术成为了另一种比较主流的方式。本文通过对 Logical Regression 逻辑回归，SVM 支持向量机和 LSTM 序列神经网络三种方式，对一系列样本数据进行测试，得出每种模型下的精确度结果，并整体对其进行分析和改进。由于本课题是工程课题，因此本文主要记录了实验过程中的相关信息和数据结果。

关键词：恶意 URL 检测，逻辑回归，SVM，LSTM

A Machine Learning System for Potential Malicious URL Detection

Aerber Zhou
Lanzhou University, School of Information Science and Engineering,Security Information
Gansu，Lanzhou 510006

Abstract: With the popularity of the Internet, more and more novel malicious URL attacks appear on the network, and the method of blacklisting mechanism can not provide complete protection. For these URL, detection technology based on machine learning classification algorithm has become another mainstream method. In this paper, we use a series of data to test through three ways of Logical Regression(LG), Support Vector Machine(SVM) and LSTM sequence neural network, and obtaine the results of each model. In the last we carry out the overall analysis and improvement. For this project topic, this article mainly record the relevant information and data results in the experiment process.

Keywords: URL Attack Detection, Logical Regression, Support Vector Machine, Long Short Term

1. 引言

近年来, 随着移动网络环境的不断完善和智能手机的普及, 互联网应用已深入渗透到生活的方方面面。CNNIC 发布的报告 [1] 显示, 截至 2016 年 6 月, 中国网民规模达 7.10 亿，互联网的普及率达到 51.7%, 超过了全球平均水平 3.1 个百分点。万维网通常是人们接入互联网的主要入口, 用户通过 URL(统一资源定位符) 直接或间接地获取互联网上的各类信息. 与此同时, 各种流氓网站、金融诈骗也藏身于此,URL 鱼龙混杂。2015 年, 据卡巴斯基安全公报统计, 大约有 196 万台受感染的主机通过网上银行窃取用户账户, 检测到的恶意对象包括脚本、漏洞和可执行文件等, 数量高达 1.2 亿, 通过恶意 URL 进行的攻击占整个网络攻击的 73%以上。可以说, 恶意 URL 是网络犯罪的基石。

恶意 URL 在广义上是指用户非自愿访问的网站地址, 这些网站内通常被植入了木马、病毒、广告等恶意代码, 这些恶意代码通过伪装成正常服务来诱导用户进行访问. 一旦进入这些恶意 URL, 用户通常会遭受广告弹窗、强制安装软件 [2] 或信息被盗等危害. 每当用户面对陌生的 URL 时都需要一个安全的评估, 因此对互联网的 URL 检测就显得十分必要.

当前一些主流应用针对恶意 URL 的检测主要依靠黑名单 [3] 机制, 如 Chrome 浏览器的 SafeBrowsing [2] 和 IE 浏览器的 SmartScreen [4]。然而, 黑名单机制仅对名单内的恶意 URL 检测有效, 不能识别名单以外的恶意 URL, 并且随着黑名单的不断增加, 检测效率会不断下降.

所以对于新出现的 URL，基于机器学习分类算法的检测技术成为了另一种比较主流的识别方式 [5–9]。机器学习检查技术识别恶意 URL 的主要流程如下：1）搜集训练样本；2）提取样本集中的 URL 异常特征并组成特征向量；3）根据样本集的特征向量训练出分类器模型；4）提取待检测 URL 特征，并将特征代入分类器模型中进行分类。

基于机器学习分类算法的恶意 URL 检测技术的检测准确率依赖于异常特征的提取和算法本身的特点, 每一种单独的分类算法都有自己擅长处理的数据和不擅长处理的数据，对于同一组数据，有些分类器擅长，有些分类器却不擅长。本项目主要是针对不同的方法进行相关的实践、分析和总结，最终提出改进和设计。

就机器学习在恶意 URL 上的检测，山石网科在 Blackhat Asia(亚洲黑帽大会) 上发表的论文 [10] “Beyond The Blacklists: Detecting Malicious URLS Through Machine Learning”，率先采用机器学习的方式进行侦查。其核心是基于大数据下已知的恶意软件样本，可以提炼出简洁的功能模型，代表许多不同的恶意软件中常用的相似性连接行为。这个模型可以用于检测有着共同的网络特征的未知恶意软件变种。

2. 目前研究现状

机器学习方法能够基于大量数据进行自动化学习和训练，已经在图像、语音、自然语言处理等方面广泛应用。

对于恶意 URL 的检测主要分为两类，而大多数转化为有监督学习问题，这就需要对数据进行标注。而其他的一些研究人员则试图以无监督的方式解决问题，例如通过异常检测技术，就不需要对数据进行标注。当可以得到标注数据时，有监督学习方法通常实现更强的泛化能力。然而在很多

时候，我们很难获得精准的标注数据。更多时候，我们可能只得到一小部分恶意 URL 和大量未标记的 URL 样本，缺乏足够可靠的负例样本。另一方面，如果我们简单地以无监督的方式解决它，那么已知恶意 URL 的标注信息就难以充分利用，可能无法达到令人满意的性能。

机器学习应用于 URL 检测存在的最大的困难就是标签数据的缺乏。尽管有大量的正常访问流量数据，但恶意 URL 稀少，且变化多样，对模型的学习和训练造成困难。

2.1 基于数理统计

基于统计学习的 web 异常检测通常需要对正常流量进行数值化的特征提取和分析。例如，URL 参数个数、参数值长度的均值和方差、参数字符分布、URL 的访问频率等等。接着，通过对大量样本进行特征分布统计，建立数学模型，进而通过统计学方法进行异常检测。

2.2 基于单分类问题

由于 web 入侵黑样本稀少，传统监督学习方法难以训练。基于白样本的异常检测，可以通过非监督或单分类模型进行样本学习，构造能够充分表达白样本的最小模型作为 Profile，实现异常检测。

2.3 基于文本分析（NLP）

Web 异常检测归根结底还是基于日志文本的分析，因而可以借鉴 NLP 中的一些方法思路，进行文本分析建模。这其中，比较成功的是基于 RNNs 的长短期记忆网络（LSTM）[11] 变体的异常检测。

3. 课题计划

3.1 课题目的

通过 Logical Regression 逻辑回归，SVM 支持向量机和 LSTM 三种方式，对一系列样本数据进行测试，得出每种模型下的结果，并对整体进行分析和改进。

由于本课题是工程课题，也参考了部分文章，[12–14] 因此本文主要记录了实验过程中的相关信息和数据，并进行分析和改进，涉及到的数理理论分析会比较少。

3.2 课题内容

3.2.1 整体流程

模型建立的整体思路如下：

1. 拿到正常请求和恶意请求的数据集。
2. 对无规律的数据集进行处理得到特征矩阵。
3. 选择某种算法模型，使用特征矩阵训练检测模型。
4. 最后计算模型的准确度，并使用模型判断未知 URL 请求是恶意的还是正常的。

本课题中采用了二分类机器学习和序列模式这两种主要方式。

其中做二分类机器学习的算法有很多，常见的有逻辑回归和支持向量机。本次课题选择这两种分类模型识别恶意 URL。序列模型的分析话主要是通过构建 LSTM RNNs 来对正常或者恶意的 URL 请求进行区分。

3.2.2 基于 Logical Regression 逻辑回归

Logistic 回归是一种广义线性回归（generalized linear model），目的是面对一个回归或者分类问题，通过特征建立代价函数，然后用优化方法迭代求解出最优的模型参数，从而学习出一个 0/1 分类模型。这个模型是将特性的线性组合作为自变量，由于自变量的取值范围是负无穷到正无穷。因此，使用 logistic 函数（或称作 sigmoid 函数）将自变量映射到 (0,1) 上，映射后的值被认为是属于 y=1 的概率。

常规步骤如下：

- 寻找 h 函数（即预测函数）
- 构造 J 函数（损失函数）
- 想办法使得 J 函数最小并求得回归参数（θ）

3.2.3 基于 SVM 支持向量机

支持向量机（Support Vector Machine，SVM，又名支持向量网络）是在分类与回归分析中分析数据的监督式学习模型。给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM 训练算法创建一个将新的实例分配给两个类别之一的模型，使其成为非概率二元线性分类器。SVM 模型是将实例表示为空间中的点，这样的映射就使得单独类别的实例可以尽可能被明显的间隔分开。然后，将实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别。

SVM 作为传统机器学习的一个非常重要的分类算法，它是一种通用的前馈网络类型，最早是由 Vladimir N.Vapnik [15] 提出，目前的版本（soft margin）是 Corinna Cortes 和 Vapnik 提出 [16] 发表。

3.2.4 基于 LSTM 序列模型

LSTM 算法全称为 Long short-term memory，是一种特定形式的 RNN（Recurrent neural network，循环神经网络），可以学习长期依赖信息。最早由 Sepp Hochreiter 和 Jürgen Schmidhuber 于 1997 年提出 [11]，并在近期被 Alex Graves [17] 进行了改良和推广。LSTM 通过刻意的设计来避免长期依赖问题。所有 RNN 都具有一种重复神经网络模块的链式的形式。在标准的 RNN 中，这个重复的模块只有一个非常简单的结构，例如一个 tanh 层（图 1）。LSTM 同样是这样的结构，但是

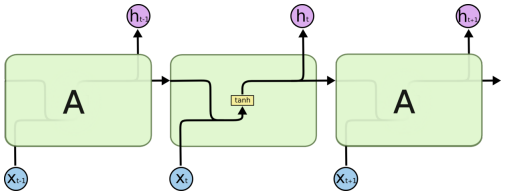

图 1: 标准 RNN 中的重复模块包含单一的层

重复的模块拥有一个不同的结构。不同于单一神经网络层，这里是有四个（图 2），以一种非常特殊的方式进行交互。LSTM 的巧妙之处在于通过增加输入门限，遗忘门限和输出门限，使得自循环的权重是变化的，这样一来在模型参数固定的情况下，不同时刻的积分尺度可以动态改变，从而避免了梯度消失或者梯度膨胀的问题。

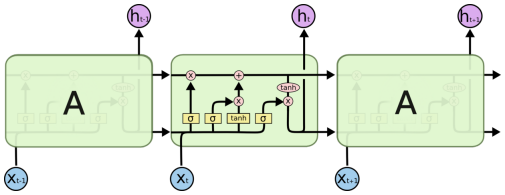

图 2: LSTM 中的重复模块包含四个交互的层

3.3 课题数据来源

由于三种测试方法的数据均来自以下来源，故在第四部分中不再赘述。对于数据，统一标记为：

- 0 = 正常请求日志样本
- 1 = 请求日志表示尝试的注入攻击

本项目一共有三大类数据集。[1] [2] [3]

图 3: 恶意 URL 部分样例

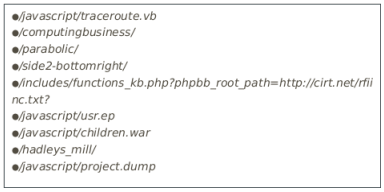

图 4: 正常访问 URL 部分样例

图 5: API 模拟 json 请求数据部分样例

[1] bad-waf.txt，good-waf.txt：来自于 Fwaf-Machine-Learning-driven-Web-Application-Firewall 项目

[2] bad1.txt, good1.txt，good2.txt: 某系统的某天的访问 URL，已进行清洗去重

[3] dev-access.csv: 通过模拟 API 给出 JSON 形式的访问请求日志

4. 基于二分类机器学习

二分类机器学习的算法有很多，常见的有逻辑回归 logical regression(lg) 和支持向量机 svm。本次课题选择这两种分类模型探究识别恶意 URL 的课题。基于二分类机器学习恶意 URL 检测算法如下：

Algorithm 1: 基于二分类机器学习恶意 URL 检测算法

```
Input: URL Text
Output: The Evaluate and Prediction
        Outputs
load data;
spilt URL text into words;
vectorize the words to get the feature
 matrix;
split the data into training set and
 validating set;
if logical regression then
    use LogisticRegression classifier;
else
    use SVM classifier;
end
train training set;
validate and evaluate the prediction;
```

4.1 数据向量化

无论是恶意请求数据集还是正常请求数据集，都是不定长的字符串列表，无法直接用逻辑回归算法对这些不规律的数据进行处理，所以，需要找到这些文本的数字特征，用来作为检测模型的输入。

在这里，本文使用 TD-IDF 方法来提取文本

的特征，并以数字矩阵的形式进行输出。TD-IDF 是一种用于资讯检索与文本挖掘的常用加权技术，被经常用于描述文本特征。TF 表示词条在某文档中出现的频率。IDF 表示的是如果包含词条的文档越少，IDF 越大，说明该词条具有很好的类别区分能力。TF-IDF 倾向于过滤掉常见的词语，保留重要的词语。

TD-IDF 输入的是文本的词语，所以需要将 URL 分词。本文选择通过长度为 N 的滑动窗口将文本分割为 N-Gram 序列。n 越大，产生的字母组合种类越多，产生的向量维度会更大，运算开销会增大，考虑到服务器的性能，这里选择 n=2。

经过分词，再 TD-IDF 处理，最终得到 URL 的特征矩阵。输出格式如下表 6：

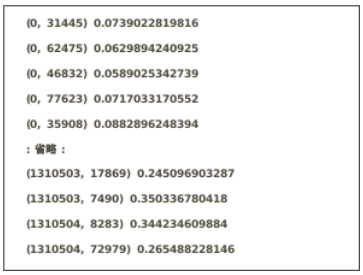

图 6: URL 向量化后的部分样例

可以看出特征矩阵的元素由 [(i,j) weight] 三个元素组成，矩阵元素 [(i,j) weight] 表示编号为 j 的词片在编号为 i 的 URL 下的 TD-IDF 值（weight）。

4.2 训练模型

得到了 URL 的特征矩阵后，我们需要将数据集分为训练集和测试集。训练集用于训练模型，测试集则是根据训练集的训练结果来评判最终的训练效果。这里我们训练集占了所有数据集合的 80%，测试集是 20%。

4.3 测试模型

经过训练之后选择一批测试数据来预测，并用 classifier 的 score 计算模型的准确度，测试数据（x_test,y_test）我们已经在上一步中分割得到。

4.4 实验结果对比

实验过程中训练均采用默认数据集：

- goodfile='good2.txt'
- badfile='bad_waf.txt'
- testfile='good1.txt'

采用 LG 和 SVM 方法得到的训练集正确率和测试集正确率见下表 1

表 1: lg 和 svm 分类器训练结果

	LG	SVM
Training Accuracy	0.9987	0.9632
Testing Accuracy	0.9974	0.8013

4.5 模型结构改进

可见 lg 分类模型对 URL 的识别相比 svm 更加精确，黑白样本整体数量是 70052 个，svm 的误差上多了 2486 个，而且训练时间也长于 lg 分类器。

对于这种情况，经过分析和查阅资料后发现，svm 对数据的异常点（离群值）比较敏感，因为其训练只支持向量，有效样本本来就不高，一旦被干扰，预测结果难以预料。同时一般输入 svm 的数据维度不宜太高，而本文训练的数据集有 4117 维，所以决定在处理前加入降维模块。

本文选择用 kmeans 降维。该算法的主要思想

是通过迭代过程把数据集划分为不同的类别，使得评价聚类性能的准则函数达到最优，从而使生成的每个聚类内部紧凑，类间独立。

表 2是对于 svm 选择 80 维和 150 维聚类，加入降维后的结果：

表 2: svm 分类器加入 Kmeans 降维训练后结果

	80 维	150 维
Training Accuracy	0.9931	0.9913
Testing Accuracy	0.9847	0.9854

可见，对数据降维后 svm 运行的速度和正确率都大大提升了，说明加入降维是有效果的。

那么对于 lg 分类器来说，是不是加入聚类之后效果也有提升呢？根据表 3来看，结果相对于原来的正确率有所下降，可见不同模型有不同的优化方法，算法的使用场景不是完全相同的。

表 3: lg 分类器加入 Kmeans 降维训练后结果

	80 维	150 维
Training Accuracy	0.9963	0.9979
Testing Accuracy	0.9949	0.9943

5. 基于 LSTM 序列模型分类

目前很多研究表明，深度学习和神经网络在图像识别和自然语言处理方面表现出众。我们可以利用神经网络的能力来处理这个分类问题。本方法是利用了 Google 的 Tensorflow 这个框架搭建相关模型，用相关数据集来判断此模型的正确率。基于 LSTM 序列模型分类的恶意 URL 检测算法如下：

Algorithm 2: 基于 LSTM 序列模型分类的恶意 URL 检测算法

Input: URL Text in JSON
Output: The Evaluate and Prediction Outputs
load data;
preprocess each URL into a sequence of characters;
map the each character of the request log text to numeric values within a word dictionary;
init a LSTM neural network model;
split the data into training set and validating set;
start train training set;
validate and evaluate the predictions;

5.1 预处理数据

本方法采用的文本数据是模拟 API 给出 JSON 形式的访问请求日志，样例请看图 5。URL 文本数据和标签（正常访问还是攻击行为）以'|' 进行切分。对 URL 文本进行预处理时由于请求日志的内容包含各种字符串、符号和数字，因此，本方法将请求日志文本中的每个字符映射到字典中的数字。相关联的数字表示该字符出现的频率。对于每一条 URL，设定固定字符长度为 1024，对于小于 1024 长度的数据部分进行头部 padding 填充，多余 1024 的直接舍弃。预处理完毕之后的数据形式如图 7：

```
array([[ 0,  0,  0, ...,  1, 17, 17],
       [ 0,  0,  0, ..., 12, 12, 17],
       [ 0,  0,  0, ..., 12, 12, 17],
       ...,
       [ 0,  0,  0, ...,  1, 17, 17],
       [ 0,  0,  0, ..., 12, 12, 17],
       [ 0,  0,  0, ..., 12, 12, 17]], dtype=int32)
```

图 7: Json URL 向量化后的部分样例

5.2 模型构建

为了快速开发神经网络模型，本文选择使用运行在 Tensorflow 之上的高级框架 Keras API。通过使用 Keras 为 Tensorflow 提供的样板设置，能够快速构建起模型原型。

模型的结构如图 8显示：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 1024, 32)          2816
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024, 32)          0
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                24832
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 27,713
Trainable params: 27,713
Non-trainable params: 0
```

图 8: LSTM 模型网络结构

LSTM 模型初始的第一层嵌入层指定了输入向量的预期维度，本文将预处理过的数据数值进行归一化，并进行 Embedding 嵌入。

第二层和第四层 Dropout 层是在训练过程中每次更新参数时按一定概率（rate）随机断开输入神经元，防止过拟合。

第三层则是模型主体部分 LSTM，LSTM 隐藏层由 64 个神经元，同时循环状态的线性变换中神经元断开比例为 50%。

最后第五层则是全连接层，并用一个 sigmoid 激活函数进行约束，以产生 01 之间的分类置信度。

5.3 训练模型

得到了 LSTM 的模型实例和处理过的数据，标签后，需要将数据集分为训练集和测试集。训练集占了所有数据集合的 75%，验证集是 25%。训练三次，每进行梯度下降时 batch 包含的样本数为 128。最后的训练结果如图 9显示：

```
Epoch 1/3
15059/15059 [==============================]
- 108s 7ms/step - loss: 0.5614 - acc: 0.7047 - val_loss: 0.2670 - val_acc: 0.9404
Epoch 2/3
15059/15059 [==============================]
- 108s 7ms/step - loss: 0.2640 - acc: 0.9218 - val_loss: 0.1697 - val_acc: 0.9562
Epoch 3/3
15059/15059 [==============================]
- 108s 7ms/step - loss: 0.2270 - acc: 0.9366 - val_loss: 0.2362 - val_acc: 0.9245
6694/6694 [==============================] - 9s 1ms/step
Model Accuracy: 92.37%
```

图 9: LSTM 模型网络训练结果

5.4 测试模型

经过训练之后，将模型对另外两个数据集进行测试，得到下表 4测试结果：

表 4: LSTM 训练后模型测试结果

	json 数据	普通 URL	FWaf 项目
Accuracy	0.9216	0.5141	0.8057

模型的结果在 json 数据格式上能过达到 92% 以上的正确率，但是用到 URL 上效果就不是那么好了，可能是数据文本形态的不同所导致的结果。

6. 总结

6.1 基于二分类的机器学习

逻辑回归和支持向量机都是常见的机器学习分类算法，从目标函数来看，区别在于 lg 采用的是 logistical loss，svm 采用的是 hinge loss。这两个损失函数的目的都是增加对分类影响较大的数据点的权重，减少与分类关系较小的数据点的权重。两者的根本目的都是一样。因此在很多实验中，两种算法的结果是很接近的。

SVM 的处理方法是只考虑支持向量，也就是和分类最相关的少数点去学习，对数据的异常点（离群值）比较敏感，而且输入 svm 的数据维度不宜太高，这就是为什么本文在第四部分中对数据进行聚类降维后 SVM 的效果会明显提升的原因。但是逻辑回归是通过非线性映射，大大减小了离分类平面较远的点的权重，相对提升了与分类最相关的数据点的权重。

逻辑回归相对来说模型更简单，更容易理解，实现起来比较方便，特别是大规模线性分类时。而 SVM 有一套结构化风险最小化的理论基础，理解和优化相对来说复杂一些。

6.2 基于 LSTM 序列模型分类器

对于 LSTM 序列模型分类器来说，由于输入是 json 格式的数据，在对 json 格式的数据分类上还是比较优良的，在 json 模拟数据集上可以达到 95% 左右的精确度，这说明这个模型确实有效。不过这些数据是在模拟数据集（Mock API）上生成，生成的 log 数据可能有一定偏向性，测试数据集可能和训练数据集风格过于一致，所以精确度很高。以至于当使用普通的纯 URL 进行测试时，精度降了很多。但是可能也是因为样本已经完全不在一个分布，数据的格式也不太相同的原因。

7. 未来工作

1. 虽然通过 Kmeans 降维之后 SVM 的检测方法有所提高，但是基于 SVM 的恶意 URL 检测很依赖于训练数据集，有必要保证原始数据集尽可能的减少噪点（离群值），以及每条数据之间尽可能的减少关联性。所以若能拿到现实生活中确定正常或者威胁的请求数据作为训练数据集的补充，来对检测模型进行优化，效果应该会更好。
2. 对于基于逻辑回归的检测方法，使用降维之后，正确率不升反降，说明降维这个方法并不是 lg 的优化策略，因为可以考虑寻找对 lg 方法进行优化的算法。
3. 在寻找数据的时候发现，白样本（正常访问 URL）的数据量远大于比黑样本（恶意 URL）的数据量，也就是出现了数据不平衡的问题（Imbalance Data），经过文献查找，这是属于 relativ imbalanced，即负例数量足够多，但相对于正例，负例相对较少。H He，EA Garcia 在 2009 年 [18] 提出相关一系列的针对数据的解决方案，例如取样方法；代价敏感方法；数据合成等，将不平衡数据通过某些方法变成平衡数据，这样可以降低正反两种数据数据量之间的差距，从而提高正确率。
4. 对于不平衡数据，我们还可以使用 Positive and Unlabeled Learning（PU Learning）[19]。

PU-Learning 是半监督学习的一种特殊情况 [20–23]，它用于解决只提供正例和未标注样本而不提供负例的问题。可以考虑通过单分类模型，学习单类样本的最小边界，边界之外的则识别为异常。简单的讲，就是放弃学习攻击的 URL 特征，不如直接学习正常 URL 的特征，拒绝所有不符合正常特征的 URL。

关于本项目

此项目已经上传 Github 公开仓库，故相关源代码不再作为附录赘述，请通过 https://github.com/NullAerber/SecurityFinalMission 自取。项目代码中有相关注释和使用说明，如有问题，可以通过 issue 提出您的意见。

参考文献

[1] Cnnic.statistical reportoninternet developmentin china,http://cnnic.com.cn/. 2016.

[2] Babak Rahbarinia, Marco Balduzzi, and Roberto Perdisci. Real-time detection of malware downloads via large-scale url-> file-> machine graph mining. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pages 783–794. ACM, 2016.

[3] Zhou Zhou, Tiecheng Song, and Yunde Jia. A high-performance url lookup engine for url filtering systems. In Communications (ICC), 2010 IEEE International Conference on, pages 1–5. IEEE, 2010.

[4] M Priya, L Sandhya, and Ciza Thomas. A static approach to detect drive-by-download attacks on webpages. In Control Communication and Computing (ICCC), 2013 International Conference on, pages 298–303. IEEE, 2013.

[5] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. IEEE Internet Computing, 11(6), 2007.

[6] Sujata Garera, Niels Provos, Monica Chew, and Aviel D Rubin. A framework for detection and measurement of phishing attacks. In Proceedings of the 2007 ACM workshop on Recurring malcode, pages 1–8. ACM, 2007.

[7] 王婷. 基于 rfe-svm 的钓鱼网页识别技术的研究. Master's thesis, 华中科技大学, 2012.

[8] 蔺亚东. 基于 url 特征的钓鱼网站检测方式. 电子测试, (2):70–72, 2014.

[9] 刘昂. 基于文本匹配的钓鱼网站检测系统的设计和实现. PhD thesis, 万方数据资源系统, 2012.

[10] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In Proceedings of the 15th ACM SIGKDD international conference on

Knowledge discovery and data mining, pages 1245–1254. ACM, 2009.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[12] 甘宏 and 潘丹. 基于 svm 和 tf-idf 的恶意 url 识别分析与研究. 计算机与现代化, (7):95–97, 2016.

[13] 汪鑫, 武杨, and 卢志刚. 基于威胁情报平台的恶意 url 检测研究. 计算机科学, 45(3):124–130, 2018.

[14] 刘健, 赵刚, and 郑运鹏. 恶意 url 多层过滤检测模型的设计与实现. 信息网络安全, (1):75–80, 2016.

[15] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. IEEE transactions on Neural Networks, 10(5):1055–1064, 1999.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

[17] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, pages 6645–6649. IEEE, 2013.

[18] Haibo He and Edwardo A Garcia. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9):1263–1284, 2009.

[19] Ya-Lin Zhang, Longfei Li, Jun Zhou, Xiaolong Li, Yujiang Liu, Yuanchao Zhang, and Zhi-Hua Zhou. Poster: A pu learning based system for potential malicious url detection. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 2599–2601. ACM, 2017.

[20] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 213–220. ACM, 2008.

[21] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In IJCAI, volume 3, pages 587–592, 2003.

[22] Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. Knowledge and Information Systems, 24(3):415–439, 2010.

[23] Marthinus C du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In Advances in neural information processing systems, pages 703–711, 2014.