计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(6):27−33 [doi: 10.15888/j.cnki.csa.006370]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于深度学习的恶意 URL 识别①

陈　康, 付华峥, 向　勇

(中国电信股份有限公司 广东研究院, 广州 510630)
通讯作者: 付华峥, E-mail: 18026262485@163.com

摘　要: 网络攻击日益成为一个严重的问题. 在这些攻击中, 恶意 URLs 经常扮演着重要角色, 并被广泛应用到各种类型的攻击, 比如钓鱼、垃圾邮件以及恶意软件中. 检测恶意链接对于阻止这些攻击具有重要意义. 多种技术被应用于恶意 URLs 的检测, 而近年来基于机器学习的方法得到越来越多的重视. 但传统的机器学习算法需要大量的特征预处理工作, 非常耗时耗力.在本文中, 我们提出了一个完全基于词法特征的检测方法. 首先, 我们训练一个 2 层的神经网络, 得到 URLs 中的字符的分布表示, 然后训练对 URL 的分布表示生成的特征图像进行分类. 在我们的试验中, 使用真实数据, 取得了精度为 0.973 和 F1 为 0.918 的结果.

关键词: 恶意 URLs; 机器学习; 词法特征; 卷积神经网络

引用格式: 陈康,付华峥,向勇.基于深度学习的恶意 URL 识别.计算机系统应用,2018,27(6):27–33. http://www.c-s-a.org.cn/1003-3254/6370.html

Malicious URL Detection Based on Deep Learning

CHEN Kang, FU Hua-Zheng, XIANG Yong

(Guangdong Research Institute, China Telecom Co. Ltd., Guangzhou 510630, China)

Abstract: Increase of cyber-attacks is now becoming a serious problem. Among these attacks, malicious URL often plays an import role. It has been widely used to mount various cyber attacks including phishing, spamming, and malware. Detection of malicious URLs is critical to thwart these attacks. Numerous techniques are developed to detect malicious URLs and machine learning techniques have been explored with increasing attention in recent years. However, traditional machine learning methods require tedious work of features preprocessing and it is very time-consuming. In this study, we propose a detection method based solely on lexical features of URLs. First, we obtain the distributed representation of characters in URLs by training a 2-layer Neural Network (NN). Then we train the Convolutional NN (CNN) to classify feature images which are generated by mapping the URL to its distributed representation. In our experience, we obtained a reasonable accuracy of 97.3% and F1 of 91.8% using the real-world data set.

Key words: malicious URLs; machine learning; distributed representation of characters; Convolutional Neural Network (CNN)

信息技术的普及极大促进了在线银行、电子商务和社交网络的发展, 人们越来越多地通过互联网完成社交、购物、资讯获取等行为, 政府也在通过互联网推行电子政务, 增强政府的透明性, 改进公共决策质量. 但同时, 互联网也成为不法分子的活跃平台, 涌现出大量的网络犯罪行为. 网络攻击者通过钓鱼网站、垃圾广告和恶意软件推广等方式非法牟利. 截至 2016 年 12 月, 中国网站数量为 482 万个, 年增长 14.1%[1]. 360 互联网安全中心截获新增钓鱼网站 196.9 万个, 同比 2015 年 (156.9 万个) 上升 25.5%; 平均每天截获新

① 基金项目: 广东省重大专项 (2015B010109005)
收稿时间: 2017-09-28; 修改时间: 2017-10-10; 采用时间: 2017-10-26; csa 在线出版时间: 2018-05-28

增 5395 个, 每小时涌现超过 225 个钓鱼网站[2]. 由公安机关与 360 安全中心联合发起的猎网平台发布《2016 年网络诈骗趋势研究报告》显示: 猎网平台共收到全国用户提交的网络诈骗举报 20 623 例, 举报总金额 1.95 亿余元, 人均损失 9471 元.

在这些攻击行为中, 有相当大的一部分是以恶意 URL 为主要手段实现的. URL, 即统一资源定位符, 是对互联网上资源的位置和访问方法的一种简洁的表示, 是互联网上标准资源的地址. 而恶意 URL 是指欺骗用户访问, 达到“执行恶意行为”或“非法窃取用户数据”目的的 URL. 攻击者通过恶意 URL 构建攻击操作的关键部分, 诱导不知情的用户访问攻击者提供的 URL, 达到其窃取用户的个人隐私信息, 例如用户的银行帐号及密码信息等, 或者下载和执行恶意程序或脚本 (例如病毒, 木马, 蠕虫等) 等攻击目的.

因此, 及时精确地检测恶意 URL, 从而有效应对大量和多种类型的网络安全攻击, 是构建网络安全解决方案中的重要一环. 为识别恶意的 URL, 近年来, 研究者已经做了较为深入的研究, 如 Liu 等人[3]利用无监督学习算法 DBSCAN 对钓鱼网页的攻击目标进行识别; Ma 等人[4]使用 Naïve Bayers 在多个公开数据集上进行检测; Huang 等人[5]提出了基于 SVM 实现了钓鱼网页识别系统.

在本文工作中, 我们提出一种基于深度学习的恶意 URL 识别模型. 本文的模型基于 URL 词法特征进行检测. 首先通过正常 URL 样本训练得到 URL 中的字符的分布表示, 将 URL 转化成二维图像, 然后通过训练 CNN 模型对二维图像进行特征抽取, 最后使用全连接层进行分类. 本文的恶意 URL 识别系统有以下两个优点:

(1) 得到 URL 中每一个字符的特征向量表示, 将字符与字符之间的编辑距离、前后顺序等信息准确包含在特征向量中;

(2) 使用 CNN 算法, 通过多个尺寸的卷积核提取 URL 中字符分布情况特征, 大大降低了传统方法中所需的特征工程带来的巨大工作量.

本文余下部分的组织为: 在第 2 节, 我们介绍当前恶意 URL 识别的相关研究进展; 第 3 节是本文的核心, 我们提出一个基于 CNN 的恶意 URL 识别模型; 在第 4 节, 我们将报告本文提出的模型在真实数据上的识别结果; 最后一节是全文工作的总结.

1 相关工作

1.1 恶意 URL 识别

目前的恶意 URL 识别工作使用的主要是黑名单启发式技术和机器学习技术.

黑名单技术是恶意网址发现算法中最传统、最经典的技术. 网页黑名单中包含已知的恶意网址列表, 通常是由具有公信力的网站根据用户举报、网页内容分析等手段生成并发布. 当用户浏览某一网址时, 基于网页黑名单的数据库就开始进行搜索. 如果这个网址在网页黑名单库中, 它就会被认为是恶意网址, 浏览器会出现警告信息; 否则认为此网址是正常网址. 在网址生成算法成熟的现在, 每天都会有大量的恶意网址出现, 黑名单技术不能够及时更新所有的恶意网址. 因此, 黑名单技术只能给与用户最低程度的保护, 并不能及时检测出恶意网站, 阻断用户对恶意网站的访问. 虽然黑名单技术有着漏判严重、更新时效性低等缺点[6], 但是其简单易用, 因此仍是许多杀毒系统常用的技术之一.

启发式算法是对黑名单技术的一种补充算法, 其主要原理是利用从恶意网址中发现的黑名单相似性规则来发现并识别恶意网页. 此算法可以依靠现有的启发式规则识别 (已有的以及部分之前未出现的) 恶意网页, 而不需要依靠黑名单的精确匹配来完成恶意网页识别. 但是, 这种方法只能为有限数量的相似恶意网页而设计, 并不能针对所有的恶意网页, 而且恶意网页要绕过此类的模糊匹配技术并不难[7]. Moshchuk 等人提出了一种更具体的启发式方法, 这些方法通过分析网页的执行动态, 比如并不寻常的过程创建、频繁的重定向等寻找恶意网页的签名[8,9]. 但是启发式算法有, 比如误报率高以及规则更新难等一些众所周知的缺点.

机器学习算法是目前研究的热点之一[10,11], 此类算法通过分析网页 URL 以及网页信息, 提取域名的重要特征表示, 并训练出一个预测模型. 目前用于恶意网页识别的机器学习算法主要分为无监督算法和有监督算法. 有监督算法也叫分类算法, 此类算法需要大量的已标注恶意/良性的网页地址作为训练集, 抽取网页特征, 然后利用现有的分类算法 (SVM、C5.0、决策树、逻辑回归等) 进行恶意网页识别. 有监督学习算法首先要对所有标注 URL 的信息进行特征提取 (域名特征、注册信息、生存时间等), 然后从中选择出能够区别恶意/良性 URL 的特征, 之后再利用分类算法进行建模分析. 此算法的准确率较高而且误报率相对较低, 但是却对

标注数据以及特征工程比较敏感, 标注数据的准确率以及选择使用的特征会严重影响算法的准确率和效率.

无监督机器学习方法又称聚类方法. 此类方法的具体分类过程主要由特征提取、聚类、簇标记和网页判别等步骤组成. 主要做法是首先将 URL 数据集划分为若干簇, 使得同一簇的数据对象之间相似度较高, 而不同簇的数据对象之间的相似度较低. 然后, 通过构造和标记数据集中的簇来区分恶意网页和良性网页. 文献[12]和[13]都提出了使用无监督机器学习识别恶意网页的具体做法, 他们都结合了域名与 IP 地址的关系来进行聚类算法的实现.

1.2 深度学习

深度学习是机器学习的一个分支. 深度学习通过学习深层非线性网络结构, 逐层训练特征, 将样本在原空间的特征表示逐步变换到新特征空间, 展现从样本集中学习数据集本质特征的强大能力. 相比于机器学习, 深度学习是唯一端到端的系统, 中间不需要人为参与, 不需要先验知[14]. 深度学习的最大好处是可以自动学习特征和抽象特征. 深度学习算法中使用最多的是卷积神经网络 (CNN) 以及循环神经网络 (RNN).

在 CNN 算法中, 相邻两层神经元之间只有部分节点相连. CNN 算法有两个很重要且特殊的部分: 卷积层和池化层. 在卷积层, 神经网络中的每一小块被深入分析从而得到抽象程度更高的特征. 一般来说. 通过卷积层的处理, 输入数据的深度会增加. 在池化层, 输入矩阵的尺寸会被有效缩小. 池化层的使用既可以加快计算速度也有防止过拟合的作用. CNN 算法经常用于图像识别领域[15], 因为它可以排除出现位置的影响有效识别图像特征[16].

RNN 算法可以充分挖掘输入数据中的时序信息以及语义信息的深度表达能力, 因此被广泛应用于语音识别、语言模型、机器翻译等领域. RNN 算法的主要用途是预测和处理序列数据. 从网络结构上讲, 循环神经网络会记忆当前序列之前的信息, 并利用之前的信息影响后面结点的输出. 也就是说, 循环神经网络的隐藏层结点之间的结点是有连接的, 隐藏层的输入不仅包括输入层的输出, 还包括上一时刻的隐藏层的输入.

2 基于卷积神经网络的恶意 URL 识别模型与实现

传统的机器学习算法非常依赖于特征, 其效果很大程度上取决于人工构造的特征的好坏. 深度神经网络能够自动提取数据特征的特性为 URL 识别提供了一种新的思路. 根据 Anh 等人[17], 正常和恶意 URL 具有不同的词法特征, 即字符出现的频率, 位置, 和前后字符的关系具有可以区分的特征, 见表 1 的恶意 URL 样例. 基于此, 我们提出一种完全基于 URL 字符串的词法特征, 利用深度神经网络实现的恶意 URL 识别算法.

表 1 恶意 URL 样例

	恶意 URL 例子
1	http://210.80.154.30/?test3/_signin.ebay.com/ebayisapidllsignin.html
2	http://21photo.cn/https://cgi3.ca.ebay.com/eBayISAPI.dllSignIn.php
3	http://vvww-baidu.com/updata.asp?f=sta&ip=192.168.122.242&mac=00-24-81-C4-29-67&pid=147186DF1E50BDDE4B2941AC293620F9&jc=[SYSTEM%20PROCESS]SYSTEMSMSS.EXECSRSS.EXEWINLOGON.EXESERVICES.EXELSASS.EXESVCHOST.EXESVCHOST.EXESVCHOST.EXESVCHOST.EXEEXPLORER.EXESVCHOST.EXEALG.EXECTFMON.EXEPYTHON.EXECONIME.EXEPYTHONW.EXECMD.EXEHOOKANAAPP.EXE147186DF1E50BDDE4B2941AC293620F9.358F55C0WMIPRVSE.EXE&time=2011-4-2811:22:10
4	http://taobao.com-item.com
5	http://168.cn.am78.nb118.com/9/mail.asp?qqnumber=%s&qqpassword=%s

识别算法分为 3 个阶段, 首先训练构成 URL 的字符表示为实数向量的形式; 其次基于第一步得到的映射表, 将 URL 转换成特征图像; 最后将特征图像输入卷积神经网络 CNN 去学习特征, 通过一个全连接层实现对 URL 的分类. 算法概述如图 1.

本算法共有两个部分: 训练部分和预测部分.

如图 1 所示, 训练流程被分为 4 个部分. 首先, 系统监控用户浏览行为过程并生成日志; 然后, 使用深度学习对日志文件进行训练得到字符的嵌入式模型; 第三步, 利用上一步得到的模型对网页 URL 进行特征转化; 最后, 使用并行的 CNN 算法训练已标注的恶意/良性 URL 特征.

在训练模型之后, 我们使用经过训练的 CNN 模型进行评估验证过程. 首先, 使用字符的嵌入式表示对日志行为数据进行特征转化; 然后, 使用训练后的 CNN 模型进行词法特征提取, 最后再使用分类输出层进行恶意概率的计算. 下面详细描述模型的训练过程.

2.1 使用语料训练 RNN 生成字符的嵌入式表示

我们将单个字符作为最小的语义单元, 一个

URL 就看成由基本单元构成的句子. 因此, 可以利用语言模型的概念对 URL 进行建模. 在训练语言模型的过程中, 得到 URL 中各字符在模型中的向量表示.

语言模型中对于词的一种主要表示方法是 one-hot 编码. 如图 2 所示, 这种方法把每个词表示为一个很长的向量. 这个向量的维度是词表大小, 其中只有一个维度的值为 1, 其它元素为 0. 值取 1 的这个维度就代表了当前的词. 这种方式如果使用稀疏方式存储, 存储效率很高, 但这种表示方法有一个很大的缺陷, 词的表示是任意的, 不能从词的表示中看出两个词之间的关系. 同时, 如果当词汇表数量大的时候, 这种方式还可能会导致维度灾难.

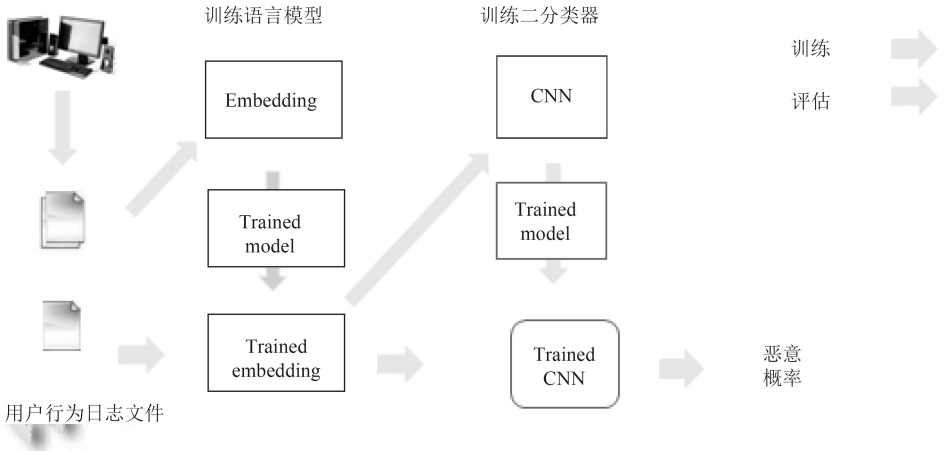

图 1 推荐模型概述

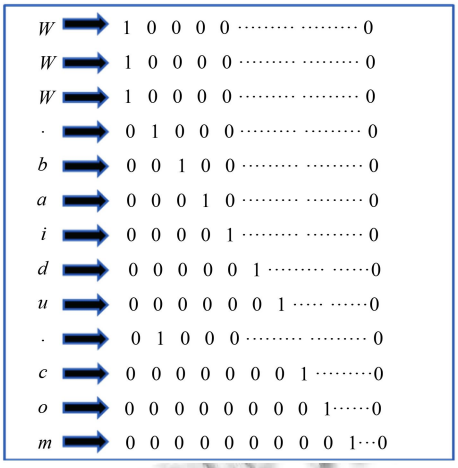

图 2 one-hot 编码

另一种更好的表示方法是分布式表示 (Distributed Representation), 分布式表示最早由 Hinton 在 1986 年提出[18]. 其基本思想是通过训练将每个词映射成 K 维实数向量 (K 一般为模型中的超参数), 不同词在 K 维空间的距离 (比如 cosine 相似度、欧氏距离等) 来表示它们之间的语义相似度. word2vec 是基于分布式表示思想的一种具体形式. 它使用了一个两层神经网络对 CBOW/Skip-Gram 模型进行训练, 在训练过程中获得一种单词在向量空间上的表示.

本文参照这个语言模型的思想, 但是针对字符而不是词进行建模. 这一步骤的主要工作是将字符映射到 K 维向量空间, 将其转为连续值的向量表示.

具体过程如下:

1) 首先从 0 开始对 URL 中出现的所有字符进行编码. 设定一个词汇表大小 v, 将所有出现的字符按出现的频率从 1 到 v–2 进行编码. 0 作为填充字符的编码, v–1 作为未知 (未出现在字符表中) 字符的编码 1;

2) 训练一个两层的神经网络模型. 构建一个[v, k]的二维向量, 将正常的 URL 作为训练样本输入. 例如 www.baidu.com, 转换为训练序列 (w, w), (w, w), (w, w), ⋯, (a, b), (a, i), ⋯, (m, o), 然后计算输出预测和它实际值的损失函数, 训练过程中更新二维向量的值;

3) 训练结束后, 得到维度为[v, k]的映射表.

2.2 将 URL 转化为特征图像

通过上述训练得到的映射表, 将 URL 转化为特征图像. 由于后续的 CNN 模型接受固定大小的图像, 我们确定一个 URL 最大长度 n, 构建一个[n, k]大小的图像. 如果 URL 长度小于 n 的, 使用 0 作为填充, 对于长度大于 n 的 URL 做截断.

转换后图像的每一行即是 URL 中的一个字符的向量表示. 如图 3 所示.

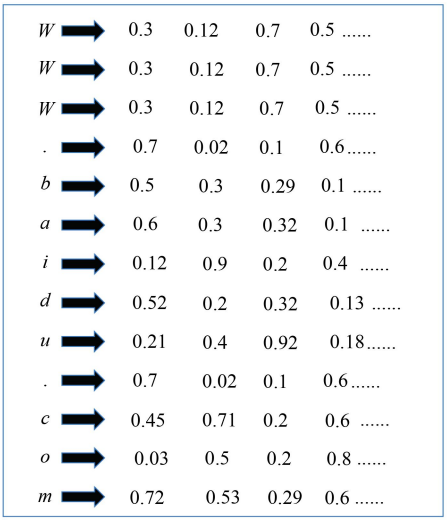

图 3 字符转换

2.3 CNN 提取特征并进行分类

构建一个 CNN 分类模型, 将上述得到的特征图像作为输入, 通过卷积层对进行特征提取, 最后通过一个全连接层进行分类, 得到输出变量-即分类结果. CNN 模型如图 4 所示.

CNN 的结构包括一个输入层, 4 个并列的卷积层以及池化层, 和一个全连接层, 最后的输出层.

上述 4 个卷积层中的每一个对应不同大小的卷积核, 高度 h 分别是 2, 3, 4, 5, 宽度为 k, 卷积核的个数取 256.

设$X_i \in R^k$是 k 维的字符向量, 对应于 URL 字符串里的第 i 个字符. 长度为 n 的 URL 字符串 (需要的时候可以进行填充或截断) 表示为$X_{1:n} = X_1 X_2 \cdots X_n$. $X_{i:(i+j)}$表示字符串$X_i X_{i+1} \cdots X_{i+j}$. 卷积操作用一个卷积核$w \in R^{hk}$应用到 h 个字符的窗口上, 生成一个新特征. 例如一个新特征 c_i 通过下式生成:

$$c_i = f(w \cdot X_{i:i+h-1} + b) \tag{1}$$

其中, $b \in R$是 bias 项, f是一个非线性函数. 该卷积核应用到 URL 字符串中的每一个可能的子串 $\{X_{1:h}, X_{2:h+1}, \cdots, X_{n-h+1:n}\}$ 上形成一个特征集:

$$c = [c_1], [c_2], \cdots, [c_{n-h+1}] \tag{2}$$

其中, $c \in R^{n-h+1}$. 非线性激活函数使用 RELU.

每一个卷积层对应一个池化层. 使用最大池化方法, 将同一个卷积核生成的特征集中最大的数值保留, 即$\hat{c} = \max\{c\}$. 将卷积层的输出尺寸转换为 1*1, 因此 256 个卷积核生成 1*256 大小的输出.

池化层的输出拼接后得到 256*4=1024 个单元作为全连接层的输入, 由于是二分类, 最后的输出层的节点是 2.

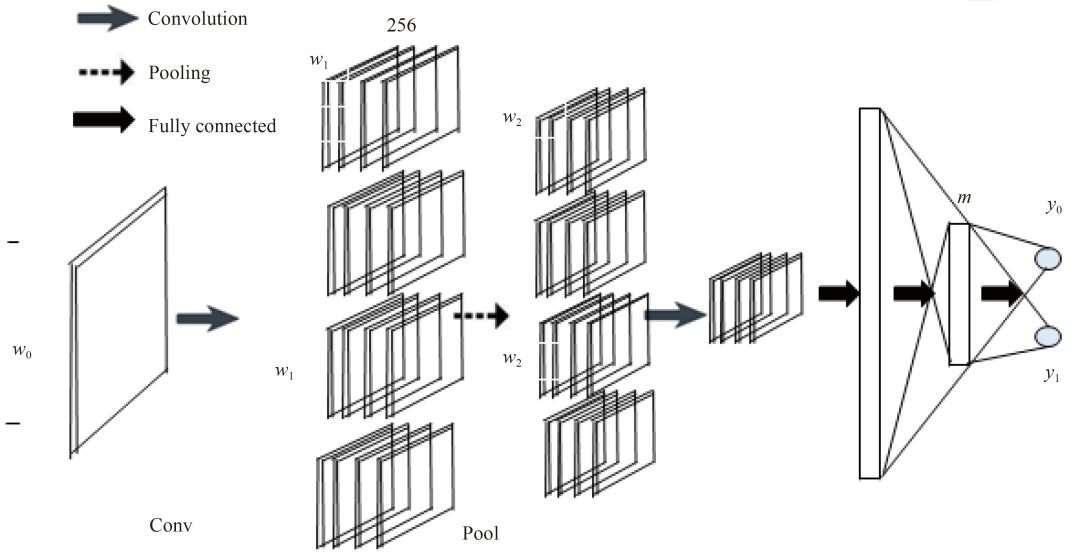

图 4 CNN 模型结构图

2.4 过拟合和 dropout

深度神经网络含有大量非线性隐含层, 使得模型表达能力非常强. 在有限的训练数据下, 导致学习了很多样本噪声的复杂关系, 而测试样本中可能并不存在这些复杂关系. 这就导致了过拟合. 本文采用了 dropout 防止过拟合.

Dropout 的意思是随机移除神经网络中的一些神经元 (隐含层和可见层), 同时包括该神经元的输入和输出. 在分类问题中, 使用 dropout 比传统的正则化方法, 泛化误差都有显著的减小.

这里在每个卷积层做完池化之后, 做一次 dropout, 防止在样本数量不大的情况下, 出现过拟合情况.

3 实验

3.1 数据与评估标准

为了获取数据, 本次研究筛选出某个月的用户浏览记录前 50 000 的 url 作为正常的 URL; 同时采用爬虫从网站 https://www.malwaredomainlist.com/mdl.php、http://www.phishtank.com/等网站中收集了 3 万多条 URL 数据作为恶意网站数据.

本次研究使用混淆矩阵以及准确率、召回率、F1 值和 ROC 曲线进行模型的评估. 在二分类算法中, 混淆矩阵的表示如表 2

表 2 混淆矩阵表示

		预测		合计
		1	0	
实际	1	True Positive (TP)	False Negative (FN)	Actual positive (TP+FN)
	0	False Postive (FP)	True Negative (TN)	Actual negative (FP+TN)
合计		Predicted postive (TP+FP)	Predicted negative (FN+TN)	TP+FP+FN+TN

根据上述矩阵中的数据, 可以得到准确率、召回率、F1 值等评价标准.

为了避免样本集中数据不均衡造成的影响, 我们同时选用了 ROC (Receiver Operating Characteristic) 曲线作为评价标准之一. ROC 也称为受试者工作特征曲线, 是一种以信息检出理论为基础, 广泛应用的数理统计方法[19]. 它根据一系列不同的阈值或者分界值, 以 TPR 为纵坐标, FPR 为横坐标绘制曲线, 曲线下面积越大, 算法精度越高.

3.2 实验结果和讨论

本次实验使用一台服务器进行, 安装了 python3.6.2, TensorFlow1.2.0, sklearn 等. 服务器操作系统是 Centos 7.2 版本, 内存为 512 G, 核数为 40.

本研究采用十折交叉验证对 80 000 多个 URL 进行分类验证. 图 5 和图 6 为算法精度和损失函数曲线和 ROC 曲线图.

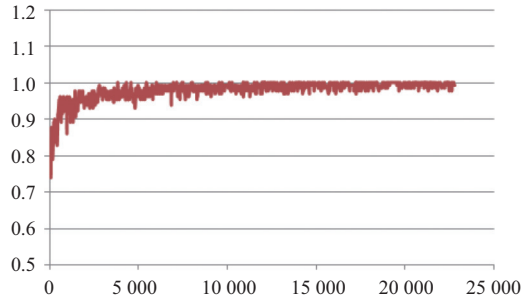

图 5 精度变化曲线图

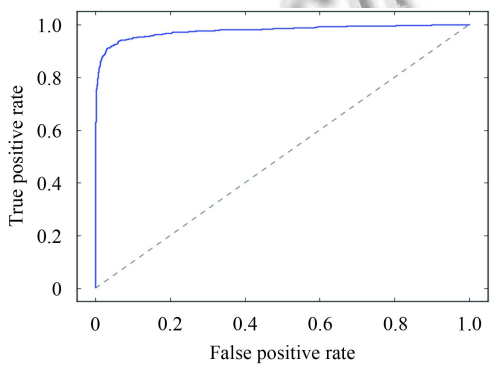

图 6 模型 ROC 曲线图

在我们的实验数据集上, 模型的准确率为 0.962、召回率为 0.879、F1 值为 0.918, 模型整体达到了很好的预测效果. 在本次试验数据的过程中, 目前测试的样本集下, 全连接层的个数会严重影响模型结果, 如果在算法的最后再增加一层全连接层, 模型效果精度将会降低 50% 左右, 因此对类似数据量的模型训练来说, 全连接层的个数至关重要. 我们使用 grid search 进行参数选择, 发现卷积核分别设置为 128 和 256, 批处理数量分别设置为 128 或者 256, 学习率设置为 0.001 时, 算法较好.

4 结束语

针对如何利用机器学习算法进行恶意域名和 URL 识别的问题, 本文提出了一种基于 URL 字符串的深度学习分类算法, 并利用 TensorFlow 进行了代码实现. 实验证明, 本文提出的恶意域名和 URL 识别分类方法, 在准确率与召回率方面都达到了较好的效果. 目前模型是二分类, 主要用于判断 URL 是否为恶意. 但是恶意 URL 种类较多, 判断出具体种类有助于进行针对性防御, 未来将进行多分类模型训练, 判断恶意 URL 种类.

参考文献

1 中国互联网络信息中心. 第 39 次《中国互联网络发展状况统计报告》. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201701/t20170122_66437.htm. [2017-01-22].

2 360 互联网安全中心. 2016 年中国互联网安全报告. http://zt.360.cn/1101061855.php?dtid=1101062514&did490278985. [2017-02-15].

3 Liu G, Qiu BT, Liu WY. Automatic detection of phishing target from phishing Webpage. Proceedings of the 20th International Conference on Pattern Recognition. Istanbul, Turkey. 2010. 4153–4156.

4 Ma J, Saul LK, Savage S, *et al*. Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA. 2009. 1245–1254.

5 Huang HJ, Qian L, Wang YJ. A SVM-based technique to detect phishing URLs. Information Technology Journal, 2012, 11(7): 921–951. [doi: 10.3923/itj.2012.921.925]

6 沙泓州, 刘庆云, 柳厅文, 等. 恶意网页识别研究综述. 计算机学报, 2016, 39(3): 529–542. [doi: 10.11897/SP.J.1016.2016.00529]

7 Sahoo D, Liu CH, Hoi SCH. Malicious URL detection using machine learning: A survey. arXiv:1701.07179. 2017.

8 Moshchuk A, Bragin T, Deville D, *et al*. Spyproxy: Execution-based detection of malicious web content. Proceedings of 16th USENIX Security Symposium. Boston, MA, USA. 2007.

9 Rieck K, Krueger T, Dewald A. Cujo: Efficient detection and prevention of drive-by-download attacks. Proceedings of the 26th Annual Computer Security Applications Conference. Austin, TX, USA. 2010. 31–39.

10 Tobiyama S, Yamaguchi Y, Shimada H, *et al*. Malware detection with deep neural network using process behavior. Proceedings of the 40th Annual Computer Software and Applications Conference. Atlanta, GA, USA. 2016. 577–582.

11 张洋, 柳厅文, 沙泓州, 等. 基于多元属性特征的恶意域名检测. 计算机应用, 2016, 36(4): 941–944, 984. [doi: 10.11772/j.issn.1001-9081.2016.04.0941]

12 张永斌, 陆寅, 张艳宁. 基于组行为特征的恶意域名检测. 计算机科学, 2013, 40(8): 146–148, 185. [doi: 10.3969/j.issn.1002-137X.2013.08.030]

13 邹福泰, 孙文杰, 谭凌霄, 等. 基于 Passive DNS 迭代聚类的恶意域名检测方法: 中国, CN106060067A. 2016-10-26.

14 胡二雷, 冯瑞. 基于深度学习的图像检索系统. 计算机系统应用, 2017, 26(3): 8–19. [doi: 10.15888/j.cnki.csa.005692]

15 Anthimopoulos M, Christodoulidis S, Ebner L, *et al*. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. IEEE Transactions on Medical Imaging, 2016, 35(5): 1207–1216. [doi: 10.1109/TMI.2016.2535865]

16 郑泽宇, 顾思宇. TensorFlow: 实战 Google 深度学习框架. 北京: 电子工业出版社. 2017. 141–145.

17 Le A, Markopoulou A, Faloutsos M. PhishDef: URL names say it all. 2011 Proceedings IEEE INFOCOM. Shanghai, China. 2011. 191–195.

18 Hinton GE. Learning distributed representations of concepts. Proceedings of the 8th Annual Conference of the Cognitive Science Society. Amherst, MA, USA. 1986. 1–12.

19 石昊苏. 基于实例与 MATLAB 的 ROC 曲线绘制比较研究. 电子设计工程, 2010, 18(9): 36–39. [doi: 10.3969/j.issn.1674-6236.2010.09.011]