基于集成学习的钓鱼网页深度检测系统①

冯 庆, 连一峰, 张颖君

(中国科学院软件研究所 可信计算与信息保障实验室, 北京 100190)

摘 要: 网络钓鱼是一种在线欺诈行为, 它利用钓鱼网页仿冒正常合法的网页, 窃取用户敏感信息从而达到非法目的. 提出了基于集成学习的钓鱼网页深度检测方法, 采用网页渲染来应对常见的页面伪装手段, 提取渲染后网页的 URL 信息特征、链接信息特征以及页面文本特征, 利用集成学习的方法, 针对不同的特征信息构造并训练不同的基础分类器模型, 最后利用分类集成策略综合多个基础分类器生成最终的结果. 针对 PhishTank 钓鱼网页的检测实验表明, 本文提出的检测方法具有较好的准确率与召回率.

关键词: 钓鱼网页; 集成学习; 深度检测; 特征提取

Depth Detection System for Phishing Web Pages Based on Ensemble Learning

FENG Qing, LIAN Yi-Feng, ZHANG Ying-Jun

(Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Phishing is a kind of online fraud that combines social engineering techniques and sophisticated attack vectors to steal the users' sensitive information to achieve the illegal purpose. In order to detect phishing web pages quickly and efficiently, this paper presents a model for depth detection of phishing web pages based on ensemble learning. The model uses page rendering to deal with common page camouflage, extract several sensitive features including URL and domain features, link and reference information, and contents of text messages; and then constructs and trains several base learning models with ensemble learning method using the features above; finally, generates the final result with base models using classification and integration method. Experiments on PhishTank indicate that the detection model this paper proposed has good accuracy and recall rate.

Key words: phishing detect; ensemble learning; feature extract

信息技术的蓬勃发展给人们的日常生活带来了便利和快捷, 人们在享受网上购物、网上银行等服务时, 也给了不法分子诸多的可乘之机. 钓鱼网站通过伪装假冒成正常合法的网站, 来诱导用户输入自己的账户密码等隐私信息, 给用户的隐私财产构成了极大的威胁[1]. 中国互联网络信息中心(CNNIC)公布的《2014 年全球中文钓鱼网站趋势分析报告》[2]显示, 我国的钓鱼网站数量呈现逐年增加的趋势, 2014 全年检测发现的钓鱼网站数量为 55063 个, 较 2012 年增长 134%, 其中仿冒淘宝网和中国工商银行网站的钓鱼网站占比 92.1%. 钓鱼网站的日益猖獗也引起了国内外专家学者的广泛关注, 如何快速有效的检测出钓鱼网页, 成为当前网络安全领域的研究热点[3].

当前钓鱼网站检测方法主要有规则过滤、页面相似度检测以及特征提取检测 3 种方法[4]. 规则过滤方法通过在用户浏览器中维持一个钓鱼网页 URL 的黑名单, 并对这个黑名单进行实时更新, 如果用户访问的 URL 在该黑名单中, 即可以阻止用户的访问. 它们通常以浏览器插件的形式存在, 比如 Chrome 浏览器自带的 Google Safe Browsing[5]、IE 浏览器中的 Microsoft

① 基金项目:国家高技术研究计划(863)(2015AA016006);国家自然科学基金(61303248,U1536106);北京市自然科学基金(4144089);信息网络安全公安部重点实验室开放课题(C15604)

收稿时间:2016-01-12;收到修改稿时间:2016-02-29 [doi:10.15888/j.cnki.csa.005360]

Phishing Filter[6]以及一些第三方的 NetCraft 和 SiteAdvisor[7,8]等. 基于规则过滤的检测方法检测速度快、准确率高, 但依赖于黑名单库的更新, 存在一定的滞后性, 而通常钓鱼网页的存活时间较短, 因此此类检测方法的召回率较低.

页面相似度检测技术就是利用恶意网页与被仿冒网页较为相似这一特性, 提取能够反映页面特征的特征向量, 通常包括文本内容特征以及视觉特征等, 采用一定的匹配算法, 计算待检测的钓鱼网页与模板库中被仿冒网页模板的相似度来进行检测. Dunlop M[9]等人提出了一种图像相似度检测的方法, 该方法首先将特征模板以及待测网页图像划分为多个子图像, 然后检测两者的相似度, 如果大于某个阈值, 则判定为钓鱼网页. 曹玖新[10]等人提出了基于嵌套 EMD(Earth Mover's Distance)的钓鱼网页检测算法, 该算法首先对网页进行图像分割, 然后提取出子图像的特征, 最后利用 EMD 算法通过计算子图像之间的相似距离来判定待测网页是否为钓鱼网页. 张卫丰[11]等人提出了一种基于匈牙利匹配算法的钓鱼网页检测方法, 该方法同时考虑文本特征、网页图片特征以及网页全局图片特征, 利用匈牙利匹配算法计算模板之间的相似度来进行判定. 页面相似度检测技术的准确率高, 但是它需要持续更新被仿冒页面的模板库.

特征提取检测技术是采用机器学习的方式对钓鱼网页进行检测, 该方法首先针对训练集中的数据提取出能表征钓鱼页面的敏感特征, 形成特征向量; 然后利用训练集的特征向量构建与训练机器学习模型; 提取待测页面的敏感特征, 利用训练形成的模型来判定是否为钓鱼页面. 机器学习在钓鱼网页检测方面得到了较好的应用, Pan[12]等人提取了 8 个钓鱼网页的 URL 特征, 训练了 SVM 分类模型, 针对共 486 个测试样本准确率为 82%. Ludl[13]等人利用了 J48 决策树分类模型, 提取了网页 URL 和 DOM 结构共 18 个特征, 采用了 4149 个合法网页和 680 个钓鱼网页进行训练, 最后得到了 96%的准确率和 83%的召回率. Xiang G[14]等人提出了 CANTINA+检测模型, 该模型提取了包括 URL 特征、页面关键词特征、PageRank、搜索排名等 15 个特征, 利用朴素贝叶斯模型进行了分类检测, 也得到了不错的检测效果. 采用机器学习的检测方式能够对未知的钓鱼网页比如 0day 钓鱼网页进行检测, 但是检测效果与训练数据集、特征提取算法以及机器学习算法都有密切的相关性, 上述几种检测模型虽然召回率较高, 但是仍有较大提升空间.

本文通过总结分析已有的研究成果, 提出了基于集成学习的钓鱼网页深度检测系统. 本系统在网页渲染、特征提取与特征选择以及模型构造方面进行了诸多改进, 并采用了集成学习的思想, 针对不同类型的特征, 构造了不同的分类学习模型, 实验表明, 本系统在钓鱼网页检测上有较高的准确率和召回率. 数据预处理方面, 首先分析了当前钓鱼页面常用的特征伪装方法, 比如用图片代替文本、构造虚假文本以及短地址等, 由此提出了页面渲染模型, 使得模型在钓鱼网页存在特征伪造的情况下依然表现良好; 在特征提取方面, 共提取了包括 URL 及域名信息、表单链接信息以及页面文本标题等在内的 16 种特征, 较为全面的包含了钓鱼网页的敏感特征; 在模型选取方面, 采用集成学习的方式, 针对不同类型特征采用不同分类模型, 比如对于 URL 特征信息、网页链接特征采用贝叶斯分类模型, 在页面内容方面采用对于高维数据表现良好的 SVM 模型, 最后采用逻辑回归的生成方式综合各个模型的结果, 使得检测效果进一步提升.

本文后续章节安排如下: 第 2 节描述系统架构, 说明模型中主要模块的功能; 第 3 节介绍网页特征的提取及表示方法; 第 4 节阐述各基础分类器的构建以及模型集成的策略; 第 5 节给出实验结果分析, 验证本文系统的有效性; 最后总结了本文工作以及对未来工作的展望.

1 系统架构

基于集成学习的钓鱼网页深度检测系统主要包含网页渲染模块、特征提取与特征选择模块、基础分类器以及分类集成模块. 系统架构如图 1 所示.

钓鱼网站为了逃避常规的检测, 通常会采用某些方式对网页进行伪装, 例如用图片替代文本、包含大量干扰检测的隐藏标签等. 本文提出了页面渲染模型, 对于一个待测 URL, 在提取特征信息之前进行去伪装处理. 钓鱼网页的伪装方式一般有如下几种:

1) 图片代替文本, 将网页关键词用图片显示, 使得特征提取检测不到关键词. 本文通过检测数据提交对象中是否包含文本, 在一定程度上能够实现对此类伪装的检测.

2) 以隐藏显示的方式构造虚假文本或者链接.

用隐藏标签显示大量与本网页无关的内容信息，用户通过浏览器看不到这些隐藏内容，但是检测系统可能会提取到大量虚假的信息. 本文研究了各类标签隐藏的方法，在提取特征前予以去除.

3) 采用js脚本动态输出页面内容. 针对此类伪装方式，本文采用网页渲染的方法对网页内容进行预处理，使得检测到的信息和浏览器看到的信息是一样的.

4) 页面自动跳转. 钓鱼网页有时为了隐藏特征，采用跳转的方式(如 js、mata refresh 等)跳转到真正的钓鱼网页. 本文同样通过网页渲染方法，获取自动跳转后的钓鱼网页.

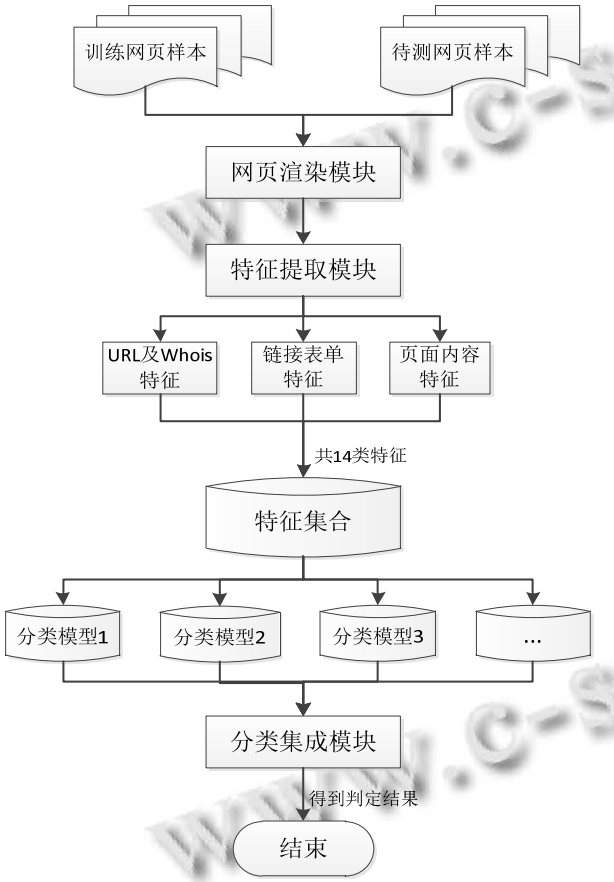

图 1 系统架构图

2 网页特征提取与表示

通过网页渲染模块对网页去伪装处理后，需要对网页进行敏感特征提取，形成特征向量. 为了能全方位的表征一个钓鱼网页，本文从URL域名、页面链接以及页面文本三方面对钓鱼网页进行特征提取. 其中URL域名特征主要包括URL中域名的段数、域名是否是IP地址、ICP备案信息、域名注册时间、URL的PageRank等；页面链接表单的特征主要包括web身份特征是否与域名特征一致、请求的外部链接比例、空链接比例、外部资源的比例、form表单中是否用图片代替文本等；页面文本特征包括页面标题信息、页面描述信息、页面关键字信息以及页面内容信息. 下面详细介绍上述特征的提取及表述方式.

2.1 URL 敏感特征的提取与表示

针对URL以及域名信息，本文提取了6个敏感特征，使用特征函数 F_i 来表示，定义 $F_{url}=(F_1,F_2,F_3,F_4,F_5,F_6)$ 为生成的URL特征向量，每个特征输出为一个整型值，表示网页中URL信息的敏感特征状态，具体的特征表述如下:

F_1: 是否使用IP地址代替域名. 此类URL的服务器可能是被远程控制的僵尸主机，F_1 的定义如下:

$$F_1=\begin{cases}1; & \text{URL中包含IP地址}\\ -1; & \text{其它}\end{cases} \tag{1}$$

F_2: 域名段数是否异常. 二级域名欺骗是当前常见的钓鱼形式，钓鱼网站为了迷惑用户，通常在其URL中包含被仿冒的网站地址，例如网址 *http://www.paypal.com.phishweb.com* 会让人误以为是paypal网站，而此类URL地址其域名段数通常较多，F_2 的定义如下:

$$F_2=\begin{cases}1; & \text{URL域名段数}\geq 5\\ -1; & \text{其它}\end{cases} \tag{2}$$

F_3: URL中是否含有异常字符. URL的异常字符包括使用“@”和“-”以及Unicode形式的字符. 例如在URL中包含“@”字符，则浏览器解析时会忽略“@”前面的字符，这是钓鱼网页迷惑用户的常用手段之一，同样“-”和Unicode也是正常网页较少采用的字符. F_3 的定义如下:

$$F_3=\begin{cases}1; & \text{URL中包含异常字符}\\ -1; & \text{其它}\end{cases} \tag{3}$$

F_4: 域名注册时间. 钓鱼网页的存活周期通常较短，网络钓鱼者为了短时间内达到非法目的，往往采用新注册的域名进行欺诈. 因此域名注册时间能够作为表征钓鱼网页的一项特征，域名注册时间可以从alexa返回的搜索结果获得，F_4 的定义如下:

$$F_4=\begin{cases}1; & \text{注册时间小于90天}\\ 0; & \text{未查询到注册信息}\\ -1; & \text{其它}\end{cases} \tag{4}$$

F_5: 网页备案信息是否一致. 网页 ICP 备案是我国为了规范网站管理而提出的一种管理策略. 该特征表示网页声明的备案信息与工信部域名备案信息是否一致, 对于英文网站, 该特征值为 0. 定义如下:

$$F_5=\begin{cases}1; & \text{有备案且信息一致}\\ 0; & \text{英文网站}\\ -1; & \text{其它}\end{cases} \tag{5}$$

F_6: 网页的 PageRank 数值. PageRank 通过网络中众多超链接的关系来确定一个页面的等级, 是 Google 搜索排名的重要依据, 网页的 PageRank 取值在 0-9 之间, 0 表示网页重要性最低, 9 表示重要性最高. F_6 的定义如下:

$$F_6=\begin{cases}\text{网页的PageRank值}\\ -1; \quad \text{未查询到PageRank}\end{cases} \tag{6}$$

2.2 页面链接特征的提取与表示

针对页面内的链接、引用表单等信息, 本文提取了能够表征钓鱼网页的 6 种敏感特征, 包括外部链接比例、空链接比例以及 Form 表单异常等. 定义 $F_{link}=(F_7,F_8,F_9,F_{10},F_{11},F_{12})$ 表示网页的链接特征. 具体的特征定义及其表述如下:

F_7: 网页身份特征是否一致. 网页身份特征是指该网页中具有某一域名最多的超链接, 由于钓鱼网页拓扑结构相对单一, 其超链接通常指向被仿网页, 因此会出现身份特征不一致的情况. F_7 的定义如下:

$$F_7=\begin{cases}1; & \text{身份特征与域名信息不一致}\\ 0; & \text{其它}\\ -1; & \text{身份特征与域名信息一致}\end{cases} \tag{7}$$

F_8: 网页空链接比例. 钓鱼网页为了短期达到目的, 制作过程往往相对简单, 因此可能包含大量的空链接. F_8 的定义如下:

$$F_8=\begin{cases}1; & L_{null}/L_{all}>=0.5\\ 0; & L_{all}=0\\ -1; & \text{其它}\end{cases} \tag{8}$$

其中 L_{null} 代表网页中空链接的数量, L_{all} 代表页面所有链接的数量.

F_9: 资源引用是否异常. 网页资源包括 DOM 结构中的<img>、<link>、<script>等. 资源引用异常特征用来衡量网页所有引用的资源中, 引用外部资源的比例. 具体的定义如下:

$$F_9=\begin{cases}1; & R_{other}/R_{all}>=0.5\\ 0; & R_{all}=0\\ -1; & \text{其它}\end{cases} \tag{9}$$

其中 R_{other} 代表网页引用外部域的资源数, R_{all} 代表网页中所有资源引用数.

F_{10}: URL 是否自动跳转. 部分钓鱼网页采用了“URL 自动跳转”以及“短地址”的机制来躲避特征检测, 一方面可以通过网页渲染进行去伪装处理, 另一方面也可以将 URL 是否自动跳转作为钓鱼页面的特征之一. F_{10} 的定义如下:

$$F_{10}=\begin{cases}1; & \text{存在自动跳转}\\ -1; & \text{不存在自动跳转}\end{cases} \tag{10}$$

F_{11}: 数据提交对象是否一致. 页面提供给用户输入数据的部分即为数据提交对象, 钓鱼网页的目的是窃取用户的敏感信息, 因此一定包含数据提交对象. 通常对于正常网页, 数据提交对象会把数据提交到同一个域内的服务器上, 而钓鱼网页为了获取用户敏感信息, 数据提交与身份会存在不一致的情况. F_{11} 的定义如下:

$$F_{11}=\begin{cases}1; & \text{空数据提交或与身份不一致}\\ -1; & \text{其它}\end{cases} \tag{11}$$

F_{12}: 是否用图片代替文本. 诸多钓鱼网页为了逃避检测, 选择用大量图片代替文本, 特征 F_{12} 主要检测网页的 Form 表单中是否有用图片代替文本, 定义如下:

$$F_{12}=\begin{cases}1; & \text{Form表单中用图片代替文本}\\ -1; & \text{其它}\end{cases} \tag{12}$$

2.3 页面文本特征提取与表示

页面文本信息包括网页的标题、描述、关键词以及内容 4 类信息, 分别用 F_{title}、F_{desc}、F_{kword}、F_{cont} 来表示提取的相应特征, 本节将以内容信息的特征提取为例, 来说明页面文本特征的提取与表示方法, 其他 3 类特征的提取及表示方法与此类似. 网页内容特征的提取包括文本提取、文本分词、特征选择、权重计算 4 个模块, 具体流程如图 2 所示.

对于训练集里的待测 URL, 通过其对应的 DOM 树结构提取文本内容信息后, 首先需要对文本内容进行分词处理, 得到一系列的单词或词语, 每一个分词结果即为一个候选特征; 然后对这些特征进行筛选, 得到能够表征出钓鱼网页特征的关键分词; 最后通过

计算待测网页中每一个分词的权重来生成待测 URL 的内容特征向量.

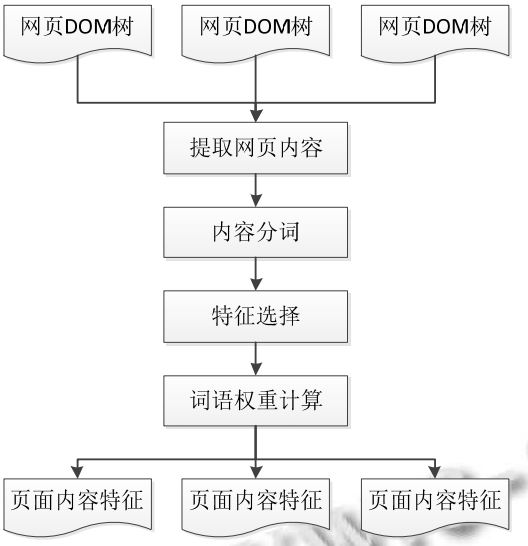

图 2 网页文本特征提取流程图

2.3.1 文本内容分词处理

对于英文页面, 其文本内容可以采用空格进行分词; 对于中文页面, 本文采用第三方开源工具“结巴分词”来进行处理. 结巴分词支持多种中文分词模式, 能够精确的将句子分开, 结巴分词的实现原理有以下三点:

1) 利用 Trie 树的结构实现高效的词图扫描, 对于句子中所有可能成词的汉字构成有向无环图;

2) 采用动态规划的思想查找最大概率的路径, 能够高效的找出基于词频的最大切分组合;

3) 对于未登录词, 使用了 Viterbi 算法, 并结合 HMM 模型进行进一步处理.

例如对于文本“淘宝网亚洲最大最安全的网上交易平台”, 采用结巴分词的全模式分词后, 结果为“淘宝, 淘宝网, 亚洲, 最大, 最, 安全, 的, 网上, 网上交易, 上交, 交易, 交易平台, 平台”.

2.3.2 文本内容特征选择

将文本内容进行分词处理后, 需要对分词结果进行筛选, 即去除那些无关钓鱼网页的特征, 从而提高分类的检测效率与准确率. 本模型采用开方检验方法 (CHI, Chi-square), CHI 特征选择算法利用了统计学中“假设检验”的基本思想: 首先假设特征词与类别直接是不相关的, 如果利用 CHI 分布计算出的检测值偏离阈值越大, 那么更有信心否定原假设, 即可以断定特征词与类别有着很高的关联度. CHI 的定义如下:

$$\chi(t_i,C_j)=\frac{N(A_{ij}D_{ij}-C_{ij}B_{ij})^2}{(A_{ij}+Ci_j)(B_{ij}+D_{ij})(A_{ij}+B_{ij})(C_{ij}+D_{ij})} \quad (13)$$

其中 $\chi(t_i,C_j)$ 表示单词 t_i 与类别 C_j 的相关度, N 为文档总数, A_{ij} 表示类别中包含单词 t_i 的文档数, B_{ij} 表示不属于类别 C_j 且包含单词 t_i 的文档数, C_{ij} 表示类别 C_j 中不包含单词 t_i 的文档数, D_{ij} 表示不属于类别 C_j 且不包含单词 t_i 的文档数. 对于同一语料而言, 文档总数 N 以及 C_j 类文档数量和非 C_j 文档的数量都是一个定值, 因此 CHI 计算公式可以化简为:

$$\chi(t_i,C_j)=\frac{(A_{ij}D_{ij}-C_{ij}B_{ij})^2}{(A_{ij}+B_{ij})(C_{ij}+D_{ij})} \quad (14)$$

因此利用 CHI 的计算公式计算出每一个单词的相关系数, 然后将相关系数从大到小排列后, 取前 M 个单词即可作为总的特征库.

2.3.3 词语权重计算

对分词结果进行特征选择后, 需要对特征库内的每一个特征计算权重, 即计算单词在该文本内容中的重要程度. 本模型采用 TF-IDF(Term Frequency-Inverse Document Frequency) 的方式来计算分词的权重. TF-IDF 是一种信息搜索和信息挖掘中常用的加权技术, 在搜索、文献分类等领域中有广泛的应用. TF-IDF 的主要思想是, 如果某一个单词在该文本内容中出现的频率很高, 并且在其它内容中出现的较少, 则可以认为该单词具有较好的区分能力, TF-IDF 的具体计算公式如下:

$$TF(X_i,j)=\frac{Count(j,X_i)}{Count(j)} \quad (15)$$

$$IDF(X_i)=\log(\frac{N}{CountFiles(X_i)}) \quad (16)$$

$$TF-IDF(X_i,j)=TF(X_i,j)*IDF(X_i) \quad (17)$$

其中, $TF(X_i,j)$ 表示文档 j 中 X_i 的词频, $Count(j,X_i)$ 表示文档 j 中单词 X_i 出现的次数, $Count(j)$ 表示文档 j 中总的单词数目; $IDF(X_i)$ 表示 X_i 的逆向频率, N 为总文档数, $CountFiles(X_i)$ 为所有包含单词 X_i 的文档数.

3 分类算法模型

本文的分类算法模型采用了集成学习的思想, 对于不同的特征类型构建了不同的分类学习模型, 集成学习通过训练多个基础分类器, 然后把这些分类器组合起来, 以达到更好的性能. 集成学习一般包含两个

步骤: 1)采用一定的基础学习算法, 根据指定的训练集或者特征集, 得到多个具有差异性的基础分类器; 2)采取某种结论生成方式, 综合每个基础分类器的输出来形成最终的分类或者预测结果.

3.1 算法模型框架

本文的基础分类器分别采用了 NBC(Native Bayes Classifier, 朴素贝叶斯分类器)模型和 SVM(Support Vector Machine, 支持向量机)两类模型, 如图 3 所示.

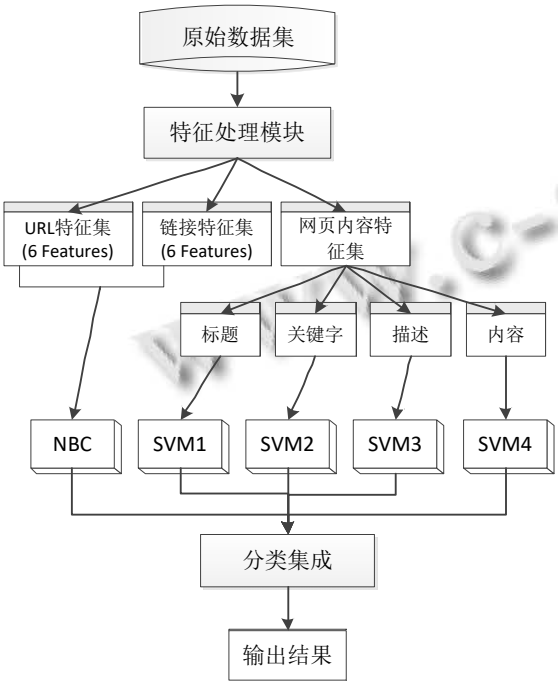

图 3　分类算法模型框架图

对于一个待测 URL, 分别提取出 URL、链接以及文本三类共 16 个特征后, 形成特征向量作为分类模型的输入. 本系统中 URL 相关的 6 种特征 $\{F_1,F_2,F_3,F_4,F_5,F_6\}$ 以及链接相关的 6 种特征 $\{F_7,F_8,F_9,F_{10},F_{11},F_{12}\}$ 用来构造 NBC 模型, 4 种文本类相关特征 F_{title} 、F_{desc} 、F_{kword} 、F_{cont} 则分别用于构建 4 个 SVM 模型. 采用这种分类策略的原因有如下三点:

1) 根据特征的不同特性, 应该采用不同的分类算法. 比如 URL 和链接相关的 12 个特征其维度较低且特征值离散, 因此可以采用在低维度下表现优异的 NBC 算法模型, NBC 算法效率较高, 模型所需的参数少, 分类结果也较稳定; 而文本相关信息比如标题、内容等, 经过分词处理后维度较高, 特征表达能力强, 而 SVM 算法对于高维度数据集有较好的处理效果, 因此对于文本相关的特征, 本文采用 SVM 算法来构建基础分类器.

2) 不同特征表达的重要性不同, 需单独使用基础分类器. 比如网页的标题信息以及关键字信息通常比文本信息更能概括页面特征, 具有更强的表征能力, 而如果将所有的文本特征构建在一起, 则无法突出标题特征的重要性.

3) 集成学习的思想是在各个基础分类器的分类结果上进行再学习, 能够大大提高模型的准确率.

3.2 NBC 基础分类模型

朴素贝叶斯模型是一种典型的生成学习模型, 通过学习已知类别的样本, 得到每一个类别下的概率分布模型, 对于一个特征集合, 得到特征集合属于各个类别下的概率, 概率最大的类别即为最终的预测结果. 朴素贝叶斯模型建立在贝叶斯定理上, 在小规模的特征集上表现良好, 运行速度快, 分类准确率高. 本文提取的 URL 和链接相关的 12 个离散特征利用朴素贝叶斯分类模型具有较高的准确率. 贝叶斯定理定义如下:

$$P(Y_k \mid X)=\frac{P(XY_k)}{P(X)}=\frac{P(Y_k)P(X \mid Y_k)}{\sum_j P(Y_j)P(X \mid Y_j)} \tag{18}$$

其中 Y_k 为分类类别, 例如是否为钓鱼网页, X 为特征集合, $P(Y_k \mid X)$ 是在已知特征集合的情况下属于类别 Y_k 的概率. 如果假设特征集 X 中所有特征之间相互独立, 公式(18)可以转换为:

$$\begin{aligned}P(X=x \mid Y=C_k)&=P(X^{(1)}=x^{(1)},...,X_{(n)}=x_{(x)} \mid Y=c_k)\\&=\prod_{j=1}^{n}P(X^{(j)}=x^{(j)} \mid Y=c_k)\end{aligned} \tag{19}$$

朴素贝叶斯模型即为得到特定特征集合下属于最可能属于的某一个分类, 即求

$$\begin{aligned}y=f(x)&=\arg\max_{c_k}P(Y=c_k \mid X=x)\\&=\arg\max_{c_k}\frac{P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)} \mid Y=c_k)}{\sum_k P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)} \mid Y=c_k)}\end{aligned} \tag{20}$$

对于特定的训练集, 公式的分母是一样的, 因此朴素贝叶斯计算公式可以化简为:

$$y=f(x)=\arg\max_{c_k}P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)} \mid Y=c_k) \tag{21}$$

3.3 SVM 分类模型

SVM 算法是 Vapnik 于 1995 年提出的基于统计学习理论(Statistical Learning Theory, SLT)的机器学习算

法. 它的基本思想是构造一个超平面, 这个决策平面能够正确的把所有的数据点分开, 并使得数据点离决策平面最远. SVM 利用了结构风险最小化原则代替了经验风险最小化, 同时采取了核函数思想, 能够将非线性的问题转化为高维线性可分的问题, 使得SVM在处理有限样本、非线性高维模式中有广泛的应用.

对于训练样本 $\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$, 其中 $x\in R^n$, $y\in\{-1,1\}$. 假设存在一个超平面 $(w^Tx)+b=0$ 能够将数据集正确分开, 则有:

$$y^{(i)}(w^Tx^{(i)}+b)\geq 1 \quad (i=1,2,...,m) \tag{22}$$

支持向量到超平面的距离为

$$\frac{|(w^Tx)+b|}{\|w\|}=\frac{1}{\|w\|} \tag{23}$$

因此可以构造最优超平面使得分类间隔最大化, 问题可以转化为如下最优化问题:

$$\begin{aligned}&\min\ \frac{1}{2}\|w\|^2\\&s.t., y^{(i)}(w^Tx^{(i)}+b)\geq 1\end{aligned} \tag{24}$$

对于线性不可分的情况引入了最优超平面的概念, 即引入松弛变量 $\zeta_i\geq 0$, 这样原目标问题可以转换为:

$$\begin{aligned}&\min\ \frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\zeta_i\\&s.t., y^{(i)}(w^Tx^{(i)}+b)\geq 1-\zeta_i\end{aligned} \tag{25}$$

其中 C 为惩罚因子, 即对于离群点的惩罚系数.

如果分类面是非线性函数的情况, 理论上需要将特征输入空间映射到一个高维空间, 实现线性可分, 并可在高维空间求出最优分类面. 由于在高维空间只需要做点积运算, 因此可以定义一种核函数 K, 将低维空间下的内积转换成高维空间. 不同的核函数能够生成不同的支撑向量机, 常见的核函数有多项式核函数、径向基核函数以及 Sigmoid 核函数等. 本文使用 SVM 模型对文本内容进行分类时, 经过交叉验证, 最终选择了径向基核函数达到了最优的效果.

3.4 结论生成策略

结论生成策略是将各个基础分类器的结果进行综合, 得到最终的分类结果. 这种综合策略可以是线性也可以是非线性的, 常用的结论生成方法有以下三种:

1) 简单投票的方法: 简单投票的方法是综合每个分类器的分类结果, 然后采用多数投票、一票否决、阈值表决等方式得到最终结果. 这种方式简单易理解, 对于基础分类器较多时有不错的准确率, 但仍有改进空间.

2) 基于贝叶斯投票的生成方法: 贝叶斯投票法依据每一个分类器在过去的表现, 利用贝叶斯定理计算出该分类器的权值, 然后进行权值投票. 在理论上贝叶斯投票法在假设空间内的情况能够具有最优的分类结果, 但是实际中不可能穷举整个假设空间, 其结果生成的稳定性较差.

3) 基于回归的生成方法: 将每一个分类器的结果看作是训练集, 采用线性回归或者逻辑回归模型进行训练, 得到每一个分类器在该假设空间内的最优解, 对于一个新的分类结果, 利用训练后的模型求出最终分类结果. 这种方法是在学习器的基础上进行再学习. 本文采用了基于逻辑回归的生成方法, 实验表明, 分类结果具有更高的准确率.

4 实验分析

4.1 评估指标

本实验从钓鱼网页检测的准确率以及召回率两个方面作为基本衡量指标. 首先定义:

TP (True Positive): 正确预测为钓鱼网站个数;

FP (False Positive): 错误预测为钓鱼网站个数;

TN (True Negative): 正确预测为合法网站个数;

FN (False Negative): 错误预测为合法网站个数;

则准确率和召回率的定义如下:

$$\text{precision(准确率)}:\ p=\frac{TP}{TP+FP}$$

$$\text{recall(召回率)}:\ r=\frac{TP}{TP+FN}$$

准确率代表检查结果的正确性, 召回率描述钓鱼网页能够被检出的概率. 这两个指标互相排斥, 如何在保证召回率的同时还能最大限度提高检测率, 是评估检测效果的关键.

4.2 实验数据与环境

在本实验中, 我们把钓鱼网页作为正例样本、正常网页作为负例样本. 所有的训练与测试数据均是爬取得到的实时数据, 钓鱼网页样本从 PhishTank 提供的黑名单中获取, 本文持续关注了 2015 年 8 月至 2015 年 12 月期间 PhishTank 提供的钓鱼网站名单, 除去非英文和中文、不可访问以及页面中没有数据提交对象

的网址，并从中随机挑选了 1500 条有效的钓鱼网页 URL 作为正例样本；合法网页的来源有 3 个方面：APWG 报告中易受攻击的合法网址、开放式网址分类目录 DMOZ、以及 Alexa 排名 TOP200 的网址，每个合法网页均爬取了一定深度下的所有网页，共 1500 条有效 URL. 具体的数据来源见表 1.

表 1　URL 测试集的来源

	钓鱼网页	合法网页
PhishTank	1500	0
APWG 报告	0	152
DMOZ	0	1024
Alexa	0	324
合计	1500	1500

本文所有工作均采用 python 语言完成，包括网页爬取、特征分析、特征选取、NBC 和 SVM 模型的使用. 其中网页渲染调用了 Selenium 第三方模块、语言检测使用了 langid 开源包、中文分词采用了第三方的结巴分词、svm 模型调用了 libsvm 开源包. 实验内容与结果分析包括：1)用 CHI 方法对文本进行特征提取，确定每种模型应选取的最优特征数；2)不同基础分类器在不同数据集下的结果分析；3)不同集成方法的比较检验.

4.3 实验结果分析

4.3.1 CHI 方法中特征个数的选择

文本信息可分为标题文本信息、关键字文本信息、描述文本信息以及内容文本信息，本文通过提取这 4 类文本特征分别构建了 4 个不同的 SVM 分类模型. SVM 分类模型虽然对较高维度的数据集表现良好，但如果不去掉诸如“的”、“和”、“the”等与是否为钓鱼网页无关的特征时，分类效率会大打折扣. 本文调用了 libsvm 工具箱中的 SVM 模型，径向基函数(RBF)作为核函数，在参数选择方面惩罚因子 C 和核参数 γ 为必备参数，这里采用 libsvm 自带的交叉验证方法来自动确定最优的参数 C 和 γ.

在确定最优特征个数前，将 1500 个钓鱼网页和 1500 个合法特征信息打乱，随机抽取 2500 条特征作为训练集，剩余 500 条作为测试集. 不同的 SVM 模型在选择不同个数的特征集时，其分类准确率如表 2 和图 4 所示.

表 2　特征个数与 SVM 模型准确率(%)关系表

	30	50	70	100	150	200	400	未提取
内容	79.9	80.7	81.5	82.4	80.8	76.2	66.8	58.4
标题	88.7	92.4	91.9	90.8	90.4	83.8	72.5	62.9
关键字	87.4	90.3	89.8	88.5	88.2	84.9	75.3	64.9
描述	83.2	87.8	88.2	86.9	86.1	83.4	72.7	65.1

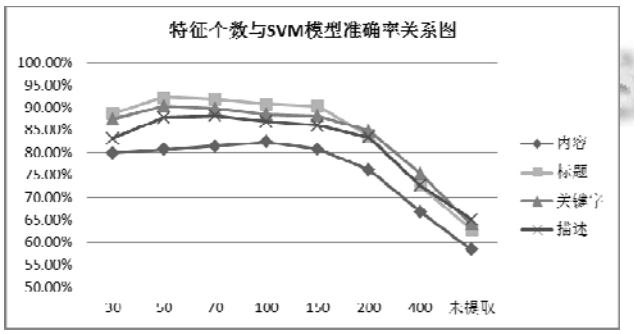

图 4　特征个数与 SVM 模型准确率关系图

可以发现当未进行特征选取时，各类 SVM 分类模型的准确率都很低，总体上标题特征构造的 SVM 模型准确率最高，其次是关键字与描述特征，最后是内容信息特征. 在特征个数选取方面，基于内容特征构造的 SVM 模型在 100 个特征时表现最好，基于描述信息的 SVM 模型在 70 个特征时表现最好，内容以及标题的 SVM 模型在 50 个特征时表现最好.

根据 CHI 方法提取的内容特征前十名如图 5 所示.

```
1       keep
2       Free
3       Paypal
4       Install
5       Sign
6       登录
7       safe
8       AOL
9       Login
10      Google
```

图 5　CHI 方法提取的 TOP10 特征值

4.3.2 基础分类器的结果分析

本文针对提取的 16 类特征构建了 5 个基础分类器，其中特征 采用朴素贝叶斯模型建模，剩余的 4 类文本

特征均采用了 SVM 模型, 分别记作: SVM_CONT, SVM_TITLE, SVM_DESC 和 SVM_KWORD. 为了全方位评估各个基础分类器的效果, 本文首先将正负例样本随机融合后, 将训练集样本分为 5 个大小不同的子集, 进一步观察各个分类器对于不同训练集大小的准确率和召回率. 测试集的大小均为 500, 样本子集的划分如表 3 所示.

表 4 以及图 6 和图 7 分别列出了各个基础分类器在不同样本子集中的准确率和召回率, 可以看到各个基础分类器的准确率和召回率都随着训练集大小的增加而增加, 其中 SVM 模型在 S2 或 S3 样本子集时, 分类效果趋于稳定, 当训练集继续增大时, 准确率和召回率增长不明显; NBC 模型在训练样本集为 1500 时趋于稳定. 各个分类器的准确率均在 82%以上, 其中 SVM_TITLE 以及 NBC 较高; 在召回率方面, NBC 模型最高, 其次由于 SVM_CONT 模型提取了大量文本信息, 召回率也较高.

表 3 样本子集的划分表

Set Name	Train Set	Test Set
S1	100	500
S2	300	500
S3	700	500
S4	1500	500
S5	2500	500

表 4 基础分类器在不同样本子集的准确率(%)和召回率(%)表

	NBC		SVM_CONT		SVM_TITLE		SVM_DESC		SVM_KWORD	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
S1	68.4	63.1	78.8	73.2	81.2	74.1	79.6	69.3	78.7	70.3
S2	72.5	69.5	81.3	78.2	86.7	76.2	84.2	73.2	84.2	73.2
S3	77.6	78.9	82.4	82.1	88	77.1	88.7	74.3	87.0	74.6
S4	86.3	90.3	82.8	83.9	89.3	77.4	89.2	74.8	88.8	75.4
S5	88.4	91.4	83	84.5	91	78.1	90	75.2	89.9	75.9

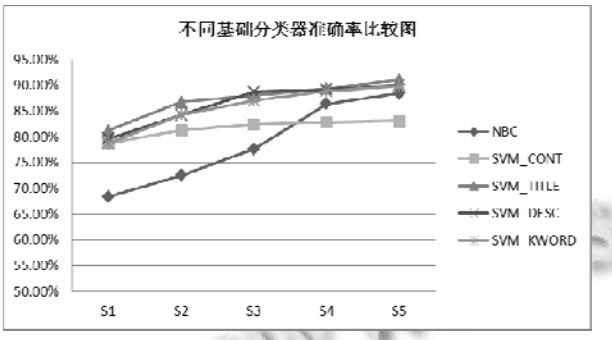

图 6 不同基础分类器的准确率比较图

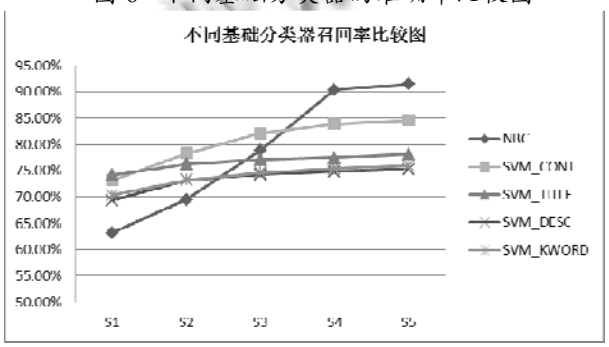

图 7 不同基础分类器的召回率比较图

4.3.3 不同集成方法的比较分析

各个基础分类器完成分类后, 需要采用一定的策略对分类结果进行综合. 各种集成方法的准确率和召回率见表 6 和图 8. 本文经过对比实验后采用了回归生成法中的逻辑回归策略, 该集成策略效果最佳, 其中准确率达到 97.2%, 召回率为 96.8%.

表 6 各种集成策略的准确率(%)和召回率(%)表

	准确率	召回率
简单投票	91.7	93.2
贝叶斯生成	96.6	95.9
线性回归	96.5	96.6
逻辑回归	97.2	96.8

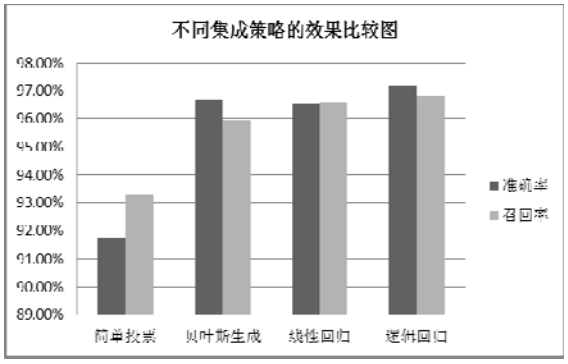

图 8 不同集成策略的效果比较图

5 结语

本文通过对大量钓鱼网页的研究, 分析了当前钓鱼网页的诸多伪装方式, 提取了包括域名、链接、页面文本在内的 16 种敏感特征, 并针对文本特征进行了特征分词筛选处理, 然后采用了集成学习的策略, 以朴素贝叶斯和支持向量机的模型为基础分类器, 集成方式采用了逻辑回归的生成方法. 通过对 PhishTank 钓鱼网页数据的检测分析, 验证了本文方法的适用性和有效性.

后续将从特征模型和分类算法两方面入手, 尝试进一步提高系统的检测效果, 并利用更大规模的数据集进行验证.

参考文献

1 中国反钓鱼网站联盟.2015 年 10 月钓鱼网站处理简报.北京: APAC,2015.

2 中国互联网络信息中心.2014 年全球中文钓鱼网站趋势分析报告.北京:CNNIC,2014.

3 CNCERT/CC. http://www.cert.org.cn.

4 Kumaraguru P, Sheng S, Acquisti A, Cranor LF, Hong J. Teaching Johnny not to fall for phish. ACM Trans. on Internet Technology, 2010, 10(2): 1–31.

5 National Consumers League. A Call for Action: Report on the National Consumers League Anti-phishing Retreat. Washington, DC, 2006: 1–57.

6 Chhabra S. Fighting Spam, Phishing and Email Fraud [Thesis]. University of California Riverside, 2005.

7 NetCraft. Netcraft Anti-Phishing tool bar. http://toolbar.netcraft.com, 2007.

8 McAfee SiteAdvisor. http://www.siteadvisor.com, 2007.

9 Dunlop M, Groat S, Shelly D. GoldPhish: Using images for content-based phishing analysis. Proc. of the 5th International Conference on Internet Monitoring and Protection. Barcelona, Spain. 2010.123–128.

10 曹玖新,毛波,罗军舟,刘波.基于嵌套 EMD 的钓鱼网页检测算法.计算机学报,2009,32(5):922–929.

11 张卫丰,周毓明,许蕾,等.基于匈牙利匹配算法的钓鱼网页检测方法.计算机学报,2010.

12 An Y, Ding X. Anomaly based web phishing page detection. Proc. of the 22nd Annual Computer Security Applications Conference on Annual Computer Security Applications Conference(ACSAC'06), Sep. 2006.

13 Ludl C, McAllister S, Kirda E, Kruegel C. On the effectiveness of techniques to detect phishing sites. Lecture Notes in Computer Science (LNCS), 2007, 4579: 20–39.

14 Xiang G, Hong J, Rose CP, Lorrie C. CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. ACM Trans. on Information & System Security, 2011, 14(2): 613–613.