

基于两层分类器的恶意网页快速检测系统研究

王正琦^{1,2}, 冯晓兵^{1,2}, 张驰^{1,2}

(1. 中国科学技术大学, 安徽 合肥 230026;

2. 中国科学院电磁空间信息重点实验室, 安徽 合肥 230026)

摘要: 针对当前传统静态恶意网页检测方案在面对海量的新增网页时面临的压力, 引入了两段式的分析检测过程, 并依次为每段检测提出相应的特征提取方案, 通过层次化使用优化的朴素贝叶斯算法和支持向量机算法, 设计并实现了一种兼顾效率和功能的恶意网页检测系统——TSMWD (two-step malicious Web page detection system)。第一层检测系统用于过滤大量的正常网页, 其特点为效率高、速度快、更新迭代容易, 真正率优先。第二层检测系统追求性能, 对于检测的准确率要求较高, 时间和资源的开销上适当放宽。实验结果表明, 该架构能够在整体检测准确率基本不变的情况下, 提高系统的检测速度, 在时间一定的情况下, 接纳更多的检测请求。

关键词: 恶意网页检测; 网络安全; 机器学习; 特征提取

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2017.00186

Study of high-speed malicious Web page detection system based on two-step classifier

WANG Zheng-qi^{1,2}, FENG Xiao-bing^{1,2}, ZHANG Chi^{1,2}

(1. University of Science and Technology of China, Hefei 230026, China;

2. Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei 230026, China)

Abstract: In view of the increasing number of new Web pages and the increasing pressure of traditional detection methods, the naive Bayesian algorithm and the support vector machine algorithm were used to design and implement a malicious Web detection system with both efficiency and function, TSMWD, two-step malicious Web page detection. The first step of detection system was mainly used to filter a large number of normal Web pages, which was characterized by high efficiency, speed, update iteration easy, real rate priority. After the former filter, due to the limited number of samples, the main pursuit of the second step was the detection rate. The experimental results show that the proposed scheme can improve the detection speed of the system under the condition that the overall detection accuracy is basically the same, and can accept more detection requests in certain time.

Key words: malicious Web page detection, network security, machine learning, feature extraction

1 引言

近年来, 社会经济、科技、文化高速发展, 互联网已逐步成为人们生活中不可或缺的重要一

环。人们正逐渐从信息时代过渡到大数据时代。网络购物、在线付款、远程会议等各式的互联网应用层出不穷, 随着互联网给人们的生活带来越来越多的便利, 更多的人希望在网络中获取自己

收稿日期: 2017-05-27; 修回日期: 2017-07-22。通信作者: 王正琦, wzqwzq@mail.ust.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61202140, No.61328208)

Foundation Item: The National Natural Science Foundation of China (No.61202140, No.61328208)

所需要的信息。因此,大量提供各式各样服务的网站与日俱增。中国互联网信息中心(CNNIC, China Internet Network Information Center) 2016年发布的第37次《中国互联网络发展状况统计报告》^[1]显示,自2014年以来,中国网站数量一直呈现明显的上升趋势。截至2016年6月,中国网站数量为4.54 M个,半年增长7.3%,如图1所示。

但是,互联网也是一把双刃剑。在方便人们生活、开拓人们视野的同时,许多不法分子也从中觅得了良机。他们或为了经济利益、或为了个人报复,借助互联网这个开放的公共平台在监管缺失的情况下进行大量的攻击行为。因此,互联网中充斥着各式各样的恶意网站、恶意软件、病毒木马等对用户个人隐私和财产安全造成巨大威胁的攻击形式,并且其传播速度和进化速度也变得越来越快,用户只要稍不留神,就会导致信息和财产的损失,并且受害的用户往往会成为攻击者的跳板,继续通过该受害用户攻击与之连接的其他用户。这种传播性极强的网络攻击行为的泛滥给当今的网络安全带来了巨大威胁,这种威胁不仅违背了网站运营商提供服务正当获利的初衷,更会使用户的操作系统程序遭到破坏,系统资源被非法控制,甚至敏感信息被盗取等,进而使用户对互联网产生一定的畏惧心理,严重影响了互联网的可信度,阻碍互联网的良性发展。

根据卡巴斯基安全实验室2016年发布的年度安全报告的统计数据^[2],恶意网页出现在了87.36%的网络攻击中并且已经超越电脑病毒成

为黑客们最常使用的获取非法收入的手段。除此以外,Google安全部门的研究表明^[3],Google有大约1.3%的搜索结果是链接到挂马网页或钓鱼网页的。由此可见,恶意网页攻击已经超过传统的恶意攻击形式成为当前网络安全领域所面临的巨大挑战,由于恶意网页一般通过脚本语言编写,具有形式灵活多变、传播速度快、影响范围广、隐蔽性高等特点,如何高效地检测出恶意网页已经成为当今网络安全领域一个亟待解决的研究课题。随着互联网的发展,每日新产生的网页数量也呈现爆炸式增长,海量的新网页不断涌入,为攻击者提供了极佳的隐蔽环境,由于受到资源和时间的限制,根据Sheng等^[4]的研究,通过传统的检测方法,47%~83%的恶意网页需要12 h后才能被发现,而大约63%的钓鱼网站在出现2 h后就已经完成首次对用户的攻击行为。因此,面对每日大批涌现出的未知网页,为了更好地保障用户的上网安全,检测系统的准确率已经不再是瓶颈。如何在有限的时间内从海量的样本中筛选出新出现的恶意样本,成了恶意网页检测面临的巨大挑战。

2 国内外研究现状

目前,国内外针对恶意网页的检测方法主要分为三大类:基于黑名单技术的识别方法、静态检测技术和动态检测技术。

2.1 基于黑名单技术的识别方法

黑名单是一份基于恶意网页的URL、IP地址、域名信息或网页的关键词信息而生成的信息列表^[5]。

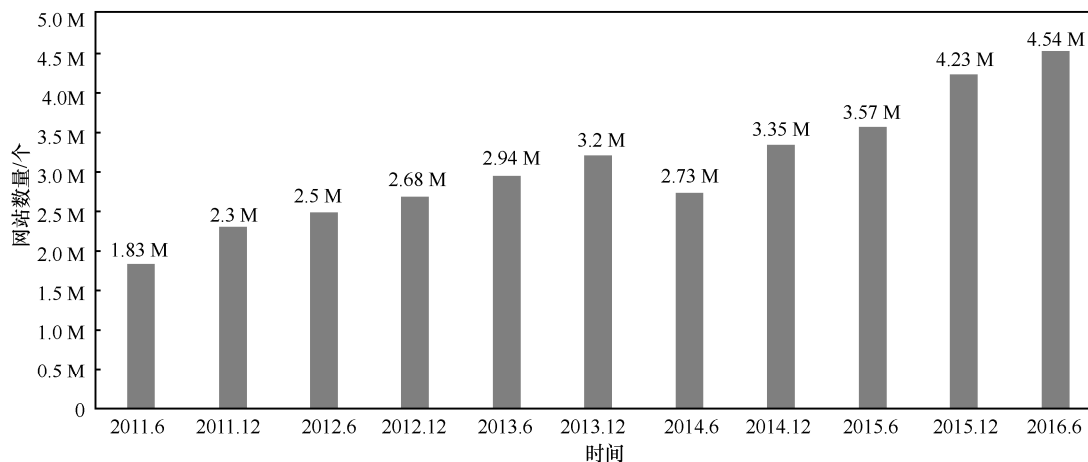


图1 中国网站数量

通过基于数据库查询的黑名单技术,人们可以准确地检测出之前已经被发现的恶意网页。每当访问一个网页时,首先在系统的数据库中查找该网页的对应信息,这样的查找仅针对网页的基本属性,而不会去解读网页本身的内容,因此只要将数据库中的内容进行遍历,一旦发现匹配成功即立刻提醒用户并进行拦截。该方案的优点十分明显,就是仅进行基础信息的匹配就可以得出结果,对于资源和时间的开销都非常小,很适用于一个成熟的恶意网页检测系统的前端,对于当前访问的网页先做一轮简单的筛选,假如该网页已经存在于黑名单中,则不必再进行后续的检测和分类。因其技术实现简单、使用方便,黑名单检测技术目前广泛应用于各类杀毒软件、安全系统解决方案中,如 Google Safe Browsing^[6]、Malware Domain List^[7]、PhishTank^[8]等项目与系统。以 Google Safe Browsing 为例,它根据 Google 提供的搜索结果,不断更新其 URL 数据库,向用户提供服务,允许用户向其提供一个 URL 地址,以判断特定的 URL 是否在其所列举的黑名单上,如果存在,则向用户发出相应的警告。PhishTank 向广大用户提供了一个可以自愿提交钓鱼网页的信息共享平台,任何用户可以向该平台提供恶意网页,经核实后收录到该开放平台的资源库中,人们可以根据该平台提供的列表过滤恶意网页,保障网络安全。

然而,黑名单检测技术有一个致命的缺陷,即该方法只能检测已经发现的恶意网页,对于刚刚新生成的恶意网页,由于还没有收录进相应列表,将不具备检测功能。Prakash 等^[9]针对该缺陷提出了一种名为 PhishNet 的改进方案,通过将已有的 URL 地址作为先验知识,对其进行分解和相似性匹配,来发现新的钓鱼网页,然而该扩展方案依然是建立在原有黑名单基础之上的扩展功能,对于某些钓鱼网页的衍生网页具有一定的检测功能,但对于新建立起的恶意网页来说,检测率仍然较低。同时黑名单检测技术由于时效性方面的缺陷,会导致大量用户在新生的恶意网页尚未被检测到之前已经受到相应的攻击。

2.2 静态检测技术

为了克服黑名单检测技术的不足,对于新出

现的网页具备一定的检测能力,研究人员们在 URL 精确匹配的基础上,提出了基于网页内容、域名信息、URL 地址等网页属性的相似性设计和启发式规则,对恶意网页进行识别和检测。静态检测技术实现的方案也有很多,可以采用基于模式匹配的方案^[10]、基于启发式规则的识别方法^[11]、基于机器学习的识别方法^[12]等。它们的本质思路都是在获得该网页相关属性信息的基础上,不在仿真环境中实际运行该段代码,而是通过与以前样本的特征进行比较,通过不同的技术手段得出最终的结论。不同于黑名单检测技术,需要完成整个字段的精确匹配,静态检测技术不需要了解恶意网页的 URL 等详细信息,可以根据网页的属性依据现有的规则进行匹配,来识别出部分目前还未被列入黑名单的网页。因为其实现相对容易,代码执行效率较高,被广泛应用于一些主流的浏览器插件中。

著名网络检测系统 Snort^[13]运用的就是基于模式匹配的静态检测技术,它将已经检测到的恶意网页代码的某些片段或关键语句,通过某种数学手段生成唯一的特征标识码并存放在 Snort 特征库中,特征码通常具备一定的长度,来确保某个特征码专门用于标识某一类特定的恶意网页,同时也存有包含通配符的特征码,用来检测更多某种类型的变种恶意代码。特征码检测方案中包含完全匹配特征码检测和模糊匹配特征码检测,前者要求待测网页中存在和特征库中的恶意特征码完全相同的片段,后者则可以通过正则表达式等模糊匹配的方式来识别网页中特定格式的代码序列,为最终判决挖掘更多的有效信息。这类方案的缺点同样在于特征库的更新具有时效性,且随着特征库的内容越来越多,模糊匹配的要求越来越高,检测所需要的资源开销也在不断增加。

基于启发式规则的静态检测技术在很多浏览器插件中应用十分广泛。该类检测方法往往建立在对于恶意网页的某些统计特征(如关键词出现的频率、特殊符号所占的比例等)是唯一的假设基础上的。Lin 等^[14]在他们的实验中展示了这一过程,他们将大量的恶意网页样本划分为多个不同的类别,每一个类别中包括一个种类的恶意网

页代码及其变形。接着对于每一个类别的恶意样本,提取其共性的特点,为之创建模板,每一个类别都对应有一个或多个不同的模板。当待测网页进入系统后,则根据其页面信息计算与各个模板之间的距离,当与某个模板的距离小于相应的阈值时,则可以认为该待测网页与该类模板相匹配,属于该类的恶意样本。Zhang 等^[15]提出了一套基于 IE 浏览器的工具插件 Cantina,通过对网页搜索的返回结果和网页中敏感关键词出现的词频等其他统计信息进行分析,建立启发式规则库对网页进行检测。启发式规则的静态检测技术的缺陷在于,面对每日大量新出现的待测样本,简单的特征分布统计和启发式规则已经无法满足需求,主要体现在以下 2 个方面:首先误检率增高,因为启发式规则的局限性和模糊匹配技术的使用,存在很多正常样本由于某些代码片段匹配某些模板,从而被误判为恶意样本;其次该方案的规则更新较难,规则的提取依赖于大量的专业知识和人工总结,因此规则更新的成本较高。

基于模式匹配和启发式规则检测的局限性,研究人员进一步提出了更加系统的基于机器学习的分类方法。作为静态检测技术的一个分支,其检测的准确性虽不能与动态检测技术同日而语,但是通过对分类器算法的改进和特征提取的优化,基于机器学习检测方案的检测率相较于其他静态检测技术已经有了很大的提高,且由于网页检测的数据量十分庞大,对检测效率要求较高,也十分符合机器学习分类算法的使用场景,可以对分类器进行不断的优化与迭代,使其检测效果

越来越好。Hou 等^[16]基于已有的恶意样本建立起一套特征提取方案,并将未知网页样本映射到已有类别中,如图 2 所示。

基于机器学习的恶意网页检测流程主要包括以下 2 个步骤。

1) 训练:通过训练样本训练出一个分类模型。每个训练样本都有一个分类的标签,即该样本是正常的还是恶意的。通过取足够数量的分类样本利用有监督学习算法,生成可以用来分类的数学模型。

2) 预测:根据待测样本所提取出的特征分布信息,经过训练好的分类器算法进行分类,判定出样本的所属类别。

Justin 等^[17]提出了一套分类特征基于 URL 和主机(host)信息的轻量级分类器,并讨论了不同机器学习算法在此场景下的表现差异。Yoo 等^[18]提出了基于误用判决模块和异常判决模块的恶意网页检测系统,其中包含 2 个基于监督与半监督机器学习算法的不同类别的分类器,与传统机器学习算法相比,该方案在正常样本的误检率略微升高的情况下,显著提升了恶意样本的检测率,但是检测所消耗的系统资源也有所上升,对于使用条件的要求更为苛刻。Canali 等^[19]提出了一个基于 URL 地址的过滤器,补充在传统的恶意网页检测系统之前,用以提升检测速度,使现有系统更加适用于大数据量的场景。但由于该过滤器仅基于 URL 信息进行过滤,导致在检测速率提高的情况下检测效率有明显下降。目前基于机器学习的检测方法所面临的问题主要有两大类:首先是特征的优化和更新问题,因为分类算法的效率很

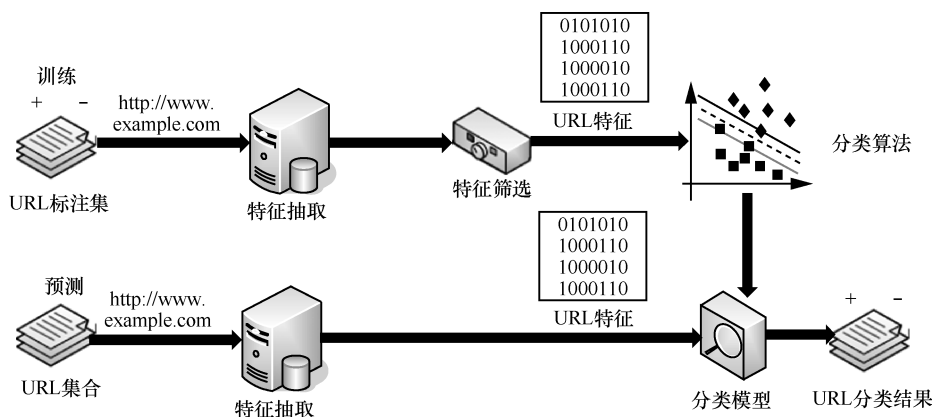


图2 基于机器学习的恶意网页检测流程

大程度上取决于特征提取的代表性，且特征提取方案具有一定的时效性，当某一批特征的检测效率过高后，会导致新一批的恶意网页进行针对性的防范，因此特征的更新与升级是面临的困难之一；第二，随着互联网的发展，恶意网页的种类越来越多，每天所要检测的未知样本数量也逐渐增多，每个样本的特征分析时间越来越长，因此检测系统的工作负荷也越来越高，如何能够降低特征提取的复杂度，减少每个样本的检测时间也是亟待解决的问题。

2.3 动态检测技术

不同于上述的静态检测技术，动态检测技术主要是通过将捕捉到的 HTML 或 JavaScript 源码在虚拟环境中运行，将虚拟机用作诱饵，对待检测网页进行访问，并在访问的过程中实时检测该网页的动态行为（如超链接到其他网页、远程下载并执行可执行文件、创建注册表等），以此来判断待测网页是否属于恶意网页。该方案的优势在于对未知网页样本，不仅通过其显示出的文本特征进行分类，而是将其放入虚拟环境中直接运行，并且诱导其在虚拟机中脱去外壳，表现出其真实目的，因此检测的准确率极高。目前最具有代表性的动态检测方案是蜜罐检测技术^[20]，该方案由 Holz 和 Koetter 在蜜罐项目中提出，用于给未知样本提供运行环境，并在蜜罐中实时检测其每一步行为。传统的蜜罐技术有低交互式服务器蜜罐和高交互式服务器蜜罐^[21]，它们实际上是将故意暴露出系统漏洞的服务器，用作诱饵以吸引攻击者对其进行攻击并分析攻击者的每一步行为。动态检测技术在恶意网页检测领域的应用保证了对可疑样本判决的准确性，但是动态检测技术的弊病也是显而易见的，无论是虚拟机脱壳引擎技术还是蜜罐检测技术，检测过程中的系统资源消耗和时间消耗都是十分巨大的，对于大样本集的使用场景，该方案的可行性较差。此外，对于具有长期潜伏特性的恶意样本，由于在虚拟机中并没有完成脱壳，其真实意图隐藏得很好，检测效果也会受到影响。

2.4 恶意网页检测评价指标

对于恶意网页检测系统的综合评价指标，主要分为两类，功能指标和性能指标。其中，功能指标

主要用于对检测系统的分类效果进行评价；而性能指标主要用于对检测系统的分类效率进行评价。

恶意网页的分类问题，是一个二值分类问题，即检测系统要将未知样本分为正常样本和恶意样本两大类，因此，要使用二值分类的评价指标对系统的分类效果进行评价，对于恶意样本的判决能力和正常样本的判决能力要分别独立进行评定。系统的目的是为了挑选出存在于大量正常样本中的恶意样本，因此将恶意网页样本作为正样本，正常网页样本作为负样本，检测系统分类结果的示意如图3所示。检测系统的功能指标主要分为以下五大类。

	+ 真实类型 -	
+ 预测类型 -	TP	FP
	FN	TN

图3 分类器功能指标示意

假设共有检测样本数量为 M ，则 $M=TP+FP+FN+TN$ ，其中每一项的含义如下。

TP (true positive): 被判决为正的正样本的数量。

FP (false positive): 被判决为正的负样本的数量。

TN (true negative): 被判决为负的负样本的数量。

FN (false negative): 被判决为负的正样本的数量。

根据这样的样本分类方式，有如下几个评价指标，来全面衡量恶意网页检测系统的功能。

准确率 (ACC, accuracy rate): 指分类正确的样本个数（包含分类正确的正样本和分类正确的负样本）占总样本个数的比重。主要用于衡量检测系统总体的准确程度。其中， T 表示所有正样本的个数， F 表示所有负样本的个数。

$$ACC = \frac{TP + FN}{T + F} \times 100\% \quad (1)$$

真正率 (TPR, true positive rate): 指分类正确的正样本占正样本总数的比重。主要用于衡量系统对恶意样本的检测能力。

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

假正率 (FPR, false positive rate): 指被误判为正样本的负样本占负样本总数的比重。主要用于衡量检测系统对于正常样本的误判水平。

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

精确率 (P, precision): 指被正确分类的正样本占有所有被分类为正样本的样本总数的比重。主要用于衡量检测系统对于恶意样本的查准率。

$$P = \frac{TP}{TP + FP} \times 100\% \quad (4)$$

召回率 (R, recall): 含义同 TPR, 主要与精确率进行对比, 衡量系统对于恶意样本的检测能力。

$$R = \frac{TP}{TP + FN} \times 100\% \quad (5)$$

对于检测系统的性能指标, 主要包括处理一个网页所需要消耗的时间、单位时间内所能处理的网页个数等。

3 TSMWD 系统的设计与实现

3.1 TSMWD 总体框架

如图 4 所示, 基于机器学习的恶意网页检测系统 (TSMWD) 包含 4 个模块: 数据采集模块、特征提取模块、数据处理模块和分类判决模块。数据采集模块主要用于训练样本和待测样本的获取, 其正样本和负样本由于数量上的差异, 获取方式有所不同。特征提取模块用于对输入的训练或待测网页内容进行解析, 该模块主要由 2 个子模块构成, 分别对应于 2 个不同的特征提取策略, 即服务于 TSMWD-I 性能优先的特征提取方案和服务于 TSMWD-II 功性优先的特征提取方案。数

据处理模块的主要功能是将特征提取模块中提取出的样本特征进行归一化、格式化等一系列处理, 在分类器的训练阶段, 该模块还负责将样本与其所属类别的标签进行一一对应, 将提取出的特征和属于该样本的类别标签以分类器需要的数据结构和数据格式进行输入。最后是分类判决模块, 在经过训练集中不同类别的样本训练后, 根据待测样本的特征分布, 给出最终的分类判决结果。在 TSMWD 中, 分类判决模块主要由 2 个部分组成, 第一部分为用于对大量未知样本进行初步筛选的 TSMWD-I, 该分类器模块的主要特点是对于恶意样本十分敏感, 恶意样本的检出率很高, 因此对该模块分类为正常的样本可以直接作为最终结果而无需进行后续的任何操作。高 TPR 的同时也意味着对正常样本有较高的误判率, 即 FPR 也较高, 因此对于 TSMWD-I 分类为恶意的样本, 本系统暂作为未知样本进行考虑, 转交 TSMWD-II 检测模块进行最终判决。因此根据不同的使用环境和检测需求, 选择不同的分类算法以及各种分类算法的参数优化是这个模块的重点。本节围绕每个模块的方案设计与实现方法进行详细讨论。

TSMWD 系统的工作流程如图 5 所示, 首先, 需要对大量的待测样本由 TSMWD-I 的特征提取方案进行特征提取, 经过数据处理模块的归一化、格式化处理后, 由分类器 TSMWD-I 进行分类判决。若判决结果为负, 即该样本为正常样本, 则检测全过程结束, 该结果就是这个待测样本的最终判决结果。若判决结果为正, 即该样本为未知样本, 则需要对当前的这个样本进行相对耗时的二次检测, 需要再次通过 TSMWD-II 的特征提取

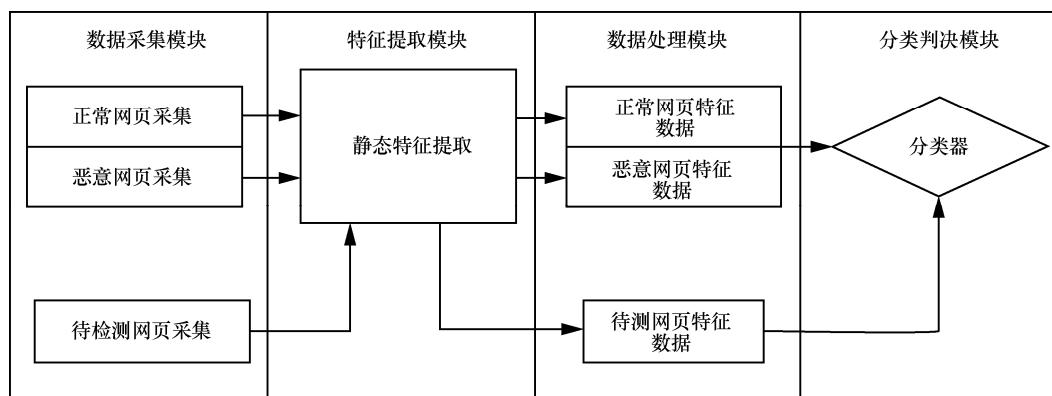


图 4 TSMWD 框架

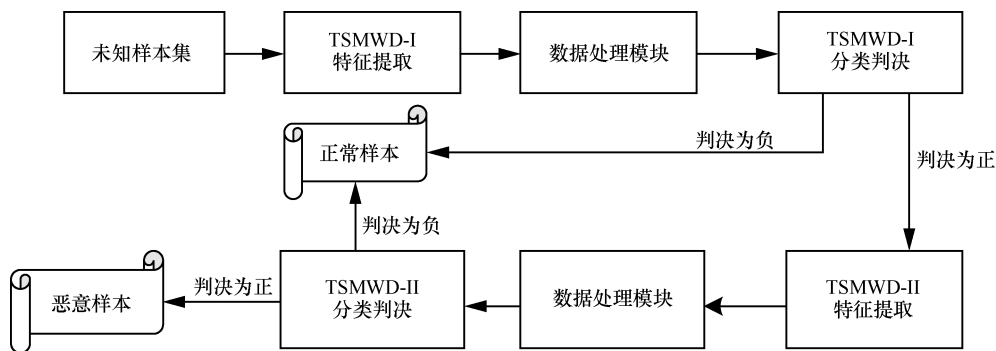


图5 TSMWD 工作流程

方案进行特征提取，该部分的特征除了包含有 TSMWD-I 的相关特征外，对于混淆、正则、敏感词出现频率等提取相对复杂和耗时的特征有了进一步补充。经过数据处理模块的归一化、格式化处理后，由分类器 TSMWD-II 给出最终的判决结果。

TSMWD 的整个工作流程大致包含 2 次侧重点不同的检测。其中，TSMWD-I 的特征提取和分类判决更加注重效率，追求速度；而 TSMWD-II 则追求精确率和召回率。对于每日新增的待测样本来说，由于正常样本的占比远高于恶意样本，因此大多数的待测样本都将会在 TSMWD-I 进行判决后就完成了整个检测过程，只有少量表现出了相关恶意样本特征的待测样本，需要进行后续的第二模块的检测。因为 TSMWD-I 的检测速率较高，能够有效过滤大多数的正常样本，因此系统总体的检测时间会明显缩减。在这样的系统架构下，为了保证检测系统的功能指标不受影响，必须最大限度地减少 TSMWD-I 造成的误判，由于 TSMWD-I 中判定为正的样本还会继续进行检测，所以负样本的误检率（FPR）（即将正常样本判定为未知样本比率）并不会最终影响整个系统的功能指标。但是对于正样本的误检率（FNR）（即将恶意样本判定为正常样本的比率），会导致待测样本在未经过 TSMWD-II 分类判决的情况下直接得出错误的判决结果，从而影响整个检测系统的准确率。为此，对 TSMWD-I 的分类算法进行了相应的校正，使其在整体最优边界的基础上向负样本一侧偏移，则其 TPR 和 FPR 会同时增高，FNR 降低，对整个系统检测率的影响达到最

低。有关分类器算法的优化和参数调整，本节给出更加全面的阐述。

3.2 TSMWD 特征提取方案

基于静态方法的特征匹配检测中必不可少的环节之一就是要能够观察并总结出恶意网页所具有的特征，所以，通过分析从 Metasploit^[22]中提取出来的不同类别的恶意样本，从多个角度和方面总结归纳恶意网页所具有的特征。在设计的特征提取方案中，有些特征属于强特征，即在正常样本和恶意样本中出现的频率有较大差距。有些特征属于弱特征，即在正常样本和恶意样本中都有出现的可能，但是基于某些统计规律，在正常样本和恶意样本中出现的频率分布有一定的区别，这样的特征也可以作为最终判决的一个参考因素。例如，对于注释的使用，正常网页中也会使用大量注释，但注释出现的位置一般存在于脚本代码开始之前或结束之后，极少出现交替穿插于脚本代码之间的情况；因此对于类似注释这样的弱特征，经过一定的修饰之后，也能够对于系统的最终分类判决产生一定的正向影响。对于不同特征权重的平衡，将通过大量样本的训练由分类器完成。由于恶意脚本的表现形式灵活多变，普通的精确字符串匹配算法无法涵盖所有形式的恶意脚本，在大量待测样本环境中的匹配效率也不高，所以利用基于有限状态机的正则表达式对恶意代码特征进行匹配就成为特征片段抓取的有效方法。根据系统设计的总体布局可知，网页源码的内容及其整体的结构布局是对一个网页进行评判的重要因素。因为本文采用静态代码检测的方式对未知网页的属性进行判断，因此在上文分

析了多种不同的 EK 攻击方式后,需要在了解其攻击原理的基础上,提炼其源码的表面特征,这些特征大致需要具有以下特点。

1) 特征之间要尽可能相互独立,将彼此间的相互作用降低到最小。

2) 特征的选取要基于正则、字符串匹配、关键字查找等相对速度较快的查询操作,而不应该涉及页面脚本代码运行后的行为数据。

3) 特征选取的覆盖面要尽可能广泛。对大量的恶意代码研究发现,恶意的代码片段不仅会出现在 JavaScript 的脚本中,即便是解释性的 HTML 或相对小众的 VBScript 中也有可能出现。

4) 特征的选取要尽量具有代表性。因为提取出的恶意网页特征,最终会用于训练分类器,所以每一个设计出的特征,都需要在恶意样本中出现的频率和在正常样本中出现的频率有一定的区别,否则如果正常样本中的出现频次和恶意样本中相近,则认为该特征的设计有一定缺陷,对于分类器的训练并不能起到正向的帮助,并且是大量误判产生的源头,要尽可能将特征在设计时就做好区分度的调研,并且对容易造成误判的敏感特征做进一步的优化和修饰。

同时,由于静态恶意代码检测的最大优点在于系统运行的速度优于动态检测,所以,在设计特征提取方案时,要考虑性能的最优化。特征提取模块的总体框架如图 6 所示, TSMWD 系统特征提取主要分为三大块,分别是 HTML 模块、JavaScript 模块和 VBScript 模块,其中 JavaScript 脚本中嵌入的恶意代码片段最为频繁,因此设计时对应的特征数量也最多。为避免网页的多次读取、相应关键数据可以先保存在内存中,以供不同的特征和方法进行

调用。最后将所有待测网页的特征写入相应文件中,供后续模块进行分析处理。

3.3 TSMWD-I 分类算法优化

TSMWD 的整个工作流程大致包含 2 次侧重点不同的检测,其中 TSMWD-I 的特征提取和分类判决更加注重检测效率,追求速度,对于每日新增的待测样本来说,由于正常样本的占比远高于恶意样本,因此大多数的待测样本都会在 TSMWD-I 进行判决后就完成整个检测过程,只有少量表现出相关恶意样本特征的待测样本需要进行后续的第二模块的检测。因为 TSMWD-I 的检测速率较高、能够有效过滤大多数的正常样本,因此系统总体的检测时间会明显缩减。在这样的系统架构下,为了保证检测系统的功能指标不受影响,必须最大限度地减少 TSMWD-I 造成的误判,由于 TSMWD-I 中判定为正的样本还会继续进行检测,所以负样本的误检率(即将正常样本判定为未知样本比率)并不会最终影响整个系统的功能指标。但是对于正样本的误检率(即将恶意样本判定为正常样本的比率),会导致待测样本在未经过 TSMWD-II 分类判决的情况下直接得出错误的判决结果,从而影响整个检测系统的准确率。

因此,对于该模块的分类,需要选择在数据量较大的情况下运算量相对较小、速率相对较快的算法,且要具备相应的矫正因子。在该使用场景下,需要牺牲 FPR,优先保证 TPR,因为对于“疑似”的恶意样本,尽管 TSMWD-I 分类器的结论更加倾向于该样本属于正常网页,但从整个系统准确率的角度出发,不能接受在第一层判决模块将恶意样本误判为正常样本。反之,将正常

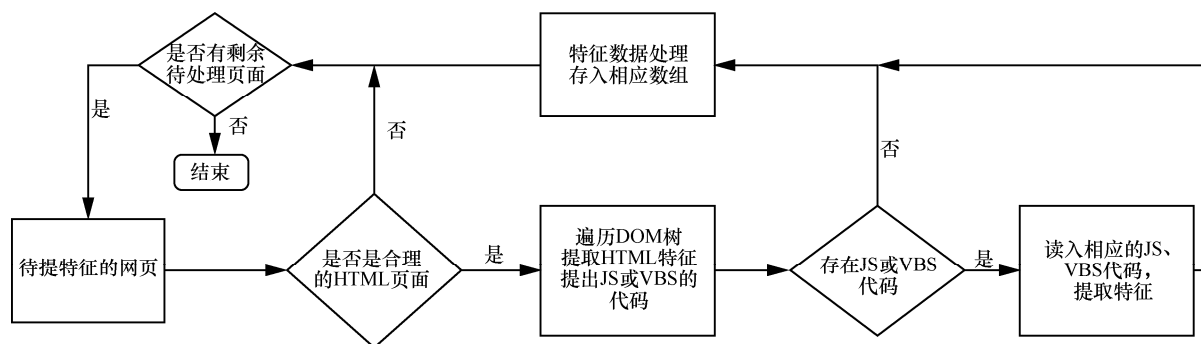


图6 特征提取模块的总体框架

样本误判为恶意样本时, 由于还要经过第二层检测系统得出最终结论, 所以这样的情况只会造成运算效率上的一点影响, 是在可接受的范围内的。因此, 对于这样的“疑似”恶意样本, 更加希望TSMWD-I能够给出未知的判决结果, 这就需要将原算法给出的最优分隔线向正常样本一侧偏移。在这样的情况下, 恶意样本的检出率会得到提高, 与此同时, 被误判为恶意样本的正常样本数量也会随之上升。

鉴于上述的使用场景, 选择基于贝叶斯定理和特征条件独立假设的朴素贝叶斯分类算法作为TSMWD-I的分类算法。不同于其他大多数分类算法(如决策树、KNN、逻辑回归、支持向量机等), 它们都属于判别方法, 也就是直接学习特征输出 Y 和特征 X 之间的关系 $Y=f(X)$, 但朴素贝叶斯算法属于生成方法, 也就是直接找出特征输出 Y 和特征 X 的联合分布 $P(X, Y)$, 然后利用贝叶斯定理求出后验概率最大的 Y , 通过计算该特征输入条件下最大概率的特征输出给出最终判决结果。朴素贝叶斯算法具有稳定的分类效率, 适合增量式训练, 尤其是数据量超出内存限制时, 可以逐批进行训练, 同时对缺失数据不敏感、算法实现相对简单、运算量较小, 非常适合文本分类的场景。

将恶意网页分类问题转化为一个二值分类题, 即任意未知样本 d 必定属于类别集合 $C=\{C_0, C_1\}$, 不妨假设 C_0 代表恶意样本类别, C_1 代表正常样本类别。未知样本 d 所具备的 n 维特征向量为 $X=\{x_1, x_2, \dots, x_n\}$, x 为特征向量 X 的某一具体实例, 则根据贝叶斯定理可知, 该样本属于类别 c 的概率为

$$P(C=c|X=x) = \frac{P(C=c)P(X=x|C=c)}{\sum_{k \in \{0,1\}} P(C=k)P(X=x|C=k)} \quad (6)$$

要想对式(6)中的所有情况进行求解, 从而得出 $P(X=x|C=c)$ 的结果, 在实际操作中是行不通的。因为特征向量 X 的维度 n 有可能会很高, 则特征向量 X 不同取值的可能性将会非常多, 且在有的可能性下样本数量极少, 存在数据量稀疏的问题。因此必须借助朴素贝叶斯的思想, 对特征向量 X 进行一个简单的假设。假设 X 中每一维

的特征都是相互独立的, 即特征与特征之间不存在任何联系。尽管在特征设计时, 有些特征间必定会存在一定的联系, 但是这种联系并不具备明显的强干扰。为了方便对 $P(X=x|C=c)$ 进行近似求解, 在此必须做出这样的特征独立性假设。由此可得

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i) (i=1,0) \quad (7)$$

将式(6)代入式(7)中可以得到拥有特征向量 X 的未知样本 d 属于类别 c 的概率转化为

$$P(C=c|X=x) = \frac{P(C=c) \prod_{i=1}^n P(X_i=x_i|C=c)}{\sum_{k \in \{0,1\}} P(C=k) \prod_{i=1}^n P(X_i=x_i|C=k)} \quad (8)$$

此时, 可以根据式(8), 通过训练样本集中不同种类的样本在不同特征分布下出现的频率, 计算出 $P(X_i|C)$ 和 $P(C)$ 。最后根据未知样本 d 在特征向量 X 的分布下, 属于不同类别的概率大小得出最后的分类结果

$$\begin{cases} X \in C_0, P(C=0|X=x) > P(C=1|X=x) \\ X \in C_1, P(C=1|X=x) > P(C=0|X=x) \end{cases} \quad (9)$$

需要注意的是, 在通过式(7)计算 $P(X_i|C)$ 的大小时, 要计算多个概率的乘积以获得该样本属于某一类别的概率。在这个计算过程中, 存在以下2个问题。

1) 如果式(7)等号右边相乘的概率中有一个概率值为0, 则最终的结果就直接为0, 这使很多种情况下得出的 $P(X_i|C)$ 的结果为0, 无法进行直观的比较。

2) 式(7)等号右侧的概率是由很多很小的数相乘组成的, 在计算 $\prod_{k=1}^n p(x_k|C_i)$ 的过程中, 由于大多数的相乘因子都非常小, 所以乘积的结果可能会发生下溢或无法得到正确的答案。

为了降低问题1)所造成的影响, 将每个特征的初始值全部置为1, 也就是说向量 X 在训练一开始就被初始化为一个 n 维的长度为1的单位向量。与此同时, 为了能够进行比较, 将其分母置为2。为了降低问题2)所造成的影响, 采用拉普拉斯平滑的解决方法^[23], 即对式(7)中的乘积取自然对数。在代数中有 $\ln(ab) = \ln a + \ln b$, 通过求

对数可以避免因乘积过小而向下溢出或者浮点数的舍入而导致的计算错误。同时，在采用自然对数进行处理时，不会对乘积之间的大小关系产生任何影响。因为函数 $f(x)$ 和函数 $\ln(f(x))$ 拥有完全相同的增减区间，并且在相同的点上取到函数的极值，所以它们的取值虽然不同，但是大小关系是完全一样的，在本文的模型中需要比较的是各个概率之间的大小关系。

根据以上的训练和预测方案，能够构建一个基于最优分界线的朴素贝叶斯算法分类器。如图7所示，在简化的二维空间中，数目相对较少的深色点为恶意样本，数目相对较多的浅色点为正常样本。当采用标准的朴素贝叶斯分类算法时，完全根据概率的大小来选择所属的类别，则图7中的黑色分隔线一定会出现在正常样本和恶意样本的交汇处。这会使整个系统的准确率相对较高，除了个别不符合特征分布规律的噪声样本外，主要的误判都集中在分隔线两侧，即当属于2个类别的概率比较接近时，容易造成分类的错误。在这种情况下，既有可能将正常样本误判为恶意样本，也有可能将恶意样本误判为正常样本。

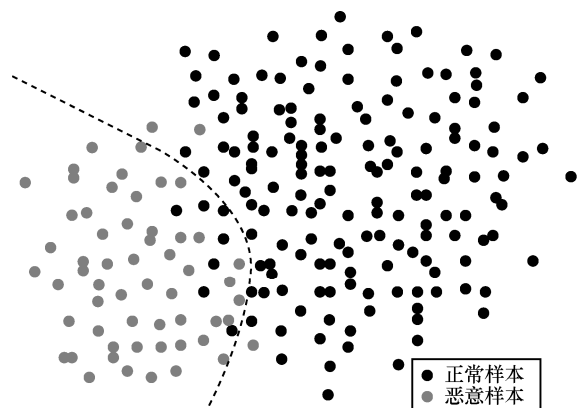


图7 最优分隔线模型

根据本节开头的描述可以知道，TSMWD-I除了追求检测效率以外，不能接受其将恶意样本误判为正常样本。反之，将正常样本误判为恶意样本时，由于还要经过第二层检测系统得出最终结论，所以这样的情况是在可接受范围内的。因此对于这样的“疑似”恶意样本，更加希望TSMWD-I能够给出未知的判决结果，这就需要原算法给出的最优分隔线向正常样本一侧偏移。在这样的情况下，恶意样本的检出率会得到提高，与此同时，被误判为恶意样本的正常样本

数量也会随之上升，但从整个TSMWD系统的角度来说，其检测准确率不会受到影响。因此需要对上述的朴素贝叶斯算法进行一些修正，使之能够满足人们的使用需求。在式(9)中引入一个矫正参数 λ ，在假设 C_0 代表恶意样本类别、 C_1 代表正常样本类别的情况下，将最终的预测模型转变为

$$\begin{cases} X \in C_0, & P(C=1|X=x) < \lambda P(C=0|X=x) \\ X \in C_1, & P(C=1|X=x) > \lambda P(C=0|X=x) \end{cases} \quad (10)$$

当 $\lambda > 1$ 时，会导致待测样本被判别为恶意样本的概率增加，图7中的分隔线向正常样本一侧偏移，如图8所示。在这样的情况下，TSMWD-I的恶意样本检出率升高，与此同时，伴随着一批正常样本被判定为恶意样本，正常样本的误检率也随之升高，符合在该场景上的分类需要。因此需要结合第二部分检测模块，通过对不同大小 λ 的尝试，在检出率和误检率间进行权衡，以找到能够最大程度地提高系统检测性能，同时又不影响系统检测准确率的 λ 。

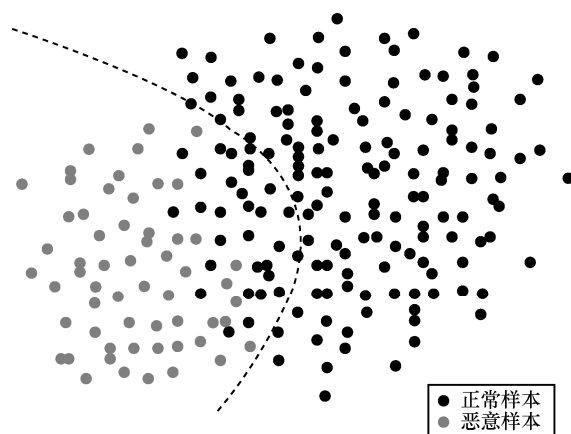


图8 矫正分隔线模型

3.4 TSMWD-II 分类算法选择

在TSMWD-I对海量待测样本进行分类的基础上，对于那些直接被分类为正常的待测样本，不需要再经过该模块的判决。对于TSMWD-I检测模块分类结果为恶意的待测样本，本系统继续视为未知样本作为TSMWD-II的输入，在TSMWD-I中提取的相关特征，已经保存在相应的中间文件中，所以可以直接通过读取与之对应的特征信息获取第一模块中既有的特征数据，该待测样本再经过TSMWD-II的特征提取模块获取一系列相对耗时的基于混淆加密、正则表达式匹配、比例计

算等行为的高级特征,生成更为全面的特征文件,由 TSMWD-II 的分类判决模块对该待测样本给出最终的判决结果。与 TSMWD-I 模块相比,TSMWD-II 的分类算法主要有以下两点不同。

1)由于 TSMWD-I 检测模块对海量正常样本的过滤,TSMWD-II 检测模块需要分类的待测样本的数量是有限的,待测样本相对较少,所以对于检测率的敏感度没有那么高,因此在 TSMWD-II 检测模块中可以不必过多地考虑算法优化对系统性能的影响,只需要关注如何提升该模块的功能指标。

2)在 TSMWD-I 中,需要保证恶意样本的检出率 TPR 足够高,这样才能将该模块对系统总体性能的影响降到最低。因此,选择了修正的朴素贝叶斯算法,在牺牲 FPR 的情况下提高了 TPR ,对于 TSMWD-I 的整体准确率会造成影响;而在 TSMWD-II 中,由于该模块给出的判决结果已经是系统的最终分类结果,因此分隔线需要选取在最优位置,保证系统总体的准确率达到最高,而不再单一地关注恶意样本的检出率,忽略正常样本的误判率。

在上述 2 个原则的基础上,本文调研了目前主流的机器学习分类算法,对不同类型的算法都进行了尝试,从准确率、真正率、假正率这 3 个方面综合比较了各分类算法的优势和缺点,从而选择出最适合该应用场景和特征提取方案的分类算法及参数模型。

在系统训练和测试的整个流程中,由于恶意样本数据集中的样本个数相对较少,没有足够多恶意的样本在保证训练分类器迭代完成后,还能提供一套完整的恶意样本测试集。为了消除噪声及数据集划分不具代表性对检测系统测试结果的影响,采用交叉验证(cross-validation)^[24]的方式对检测系统进行训练与测试,其主要思想是将数据集随机分成 k 个同样大小的子集,依次选取其中的一个子集作为测试集,其余 $k-1$ 个集合作为训练集,经过 k 次训练和测试后,将 k 次结果的平均值作为检测系统最终的实验数据。依次对 KNN 算法、决策树算法(C4.5)、分类回归树(CART, classification and regression tree)算法和支持向量机(SVM, support vector machine)算法

进行了十字交叉验证,其中线性核函数的 SVM 算法表现最好,将其用于 TSMWD-II 检测模块的分类算法。对于该模块不同分类算法实验结果的对比和整个 TSMWD 系统的相关性能指标和功能指标与已有系统的对比及其应用场景,将在第 4 节系统实验评测中进行详细分析。

4 系统实验测评

4.1 实验环境

1) 硬件环境

CPU: Intel(R) Core i3-4130;

内存: DDR3 12 G;

硬盘: 1 T, 7 200 转/秒。

2) 开发环境

开发操作系统: Windows 7 旗舰版(64 bit);

机器学习开源框架: Scikit-learn、LibSvm;

开发语言及工具: C/C++、Python2.7.14、STL 标准模板库、boost C++库、Visual Stidio2013、Pycharm、VMware、QT Creator 5.5。

4.2 TSMWD-I 实验结果

3.3 节提出了加入矫正因子 λ 的朴素贝叶斯分类算法,在 $\lambda > 1$ 的情况下,对于 λ 取不同的值,分隔线偏向正常样本一侧的程度不同,可以根据 TPR 与 FPR 的权衡选择最适合当前需求的参数值。对已经采集到的 1 428 个恶意样本与 30 000 个正常样本,采用 10 次交叉验证的方法,对于每个设定的 λ 进行 10 次实验,将 142 个恶意样本和 3 000 个正常样本作为测试集,1 286 个恶意样本和 18 000 个正常样本作为训练集,经过 10 次实验后,将实验结果进行平均。如 2.4 节中的描述,从系统检测的准确率、真正率、假正率 3 个方面对系统的整体检测能力进行评估。实验对比了不同 λ 取值下,TSMD-I 的检测能力,结果如表 1 所示。

表 1 不同 λ 取值下 TSMWD-I 检测能力对比

λ	ACC	TPR	FPR
1	78.30%	76.10%	14.20%
1.2	71.30%	85.90%	21.60%
1.4	62.90%	92.70%	31.50%
1.6	53.50%	96.30%	42.20%

通过表 1 可以看到, 当 $\lambda=1$ 时, 此时的 TSMWD-I 的分类算法就是传统的未经修正的朴素贝叶斯算法, 其准确率为 78.30%, 但是对于恶意样本的检出率只有 76.10%, 这意味着如果直接使用未经修正的朴素贝叶斯算法作为第一层检测系统的分类算法, 将会有 23.90% 的恶意样本在 TSMWD-I 对正常样本进行过滤时就已经被误判, 即便 TSMWD-II 的检测性能再强大, 也无法改变整个系统检测性能不佳的事实。随着矫正因子 λ 的增加, 分隔曲线向正常样本一侧偏移, 对于恶意样本的检测成功率也越来越高, 即 TPR 不断上升, 但同时也有更多的正常样本被误判为恶意样本, 即 FPR 也随之上升; 由于正常样本的数目相对庞大, 所以模块整体的准确率逐渐下降。但正如 3.3 节中描述的那样, 整体的准确率并不是该模块追求的目标, TSMWD-I 最重要的职责是为后续检测系统过滤掉大部分的正常样本, 同时避免对恶意样本的误判。因此注意到, 当 $\lambda=1.6$ 时, 恶意样本的真正率已经高达 96.30%, 这意味着只有 3.7% 的恶意样本会因为第一层检测系统而被误判。同时, 假正率也已经高达 42.20%, 这意味着仅有 57.80% 的正常样本会被第一模块快速过

滤, 有 42.20% 的正常样本依旧会被当作未知样本交由后续检测系统进行判决, 在这样的情况下, 对于整个系统而言, 速度的提升幅度受到了限制, 因为有近一半的待测样本最终还是需要被检测相对耗时的第二模块处理。本文对更多的 λ 取值进行了实验, 并根据实验结果做出了在不同 λ 取值下, TPR 和 FPR 之间相互权衡的经验曲线, 如图 9 所示。随着 TPR 的升高, 恶意样本的漏判率会降低, 系统整体的检测性能会变好; 但与此同时, FPR 会随之升高, 意味着有更多的正常样本不能被过滤, 需要由第二模块给出最终判决结果, 系统整体的检测时间会变长。

因此, 选择合适的 λ 取值, 也是系统整体对检测效率和检测速度之间的平衡, 是功能指标与性能指标之间的平衡。4.3 节将重点介绍 TSMWD-II 的分类器算法的选择, 然后将 2 个模块相结合, 对于 TSMWD 检测系统整体的检测速度和检测性能在不同的 λ 取值下给出更为全面的评估和测试。

4.3 TSMWD-II 实验结果

在 TSMWD 的第二层检测模块中, 分别对

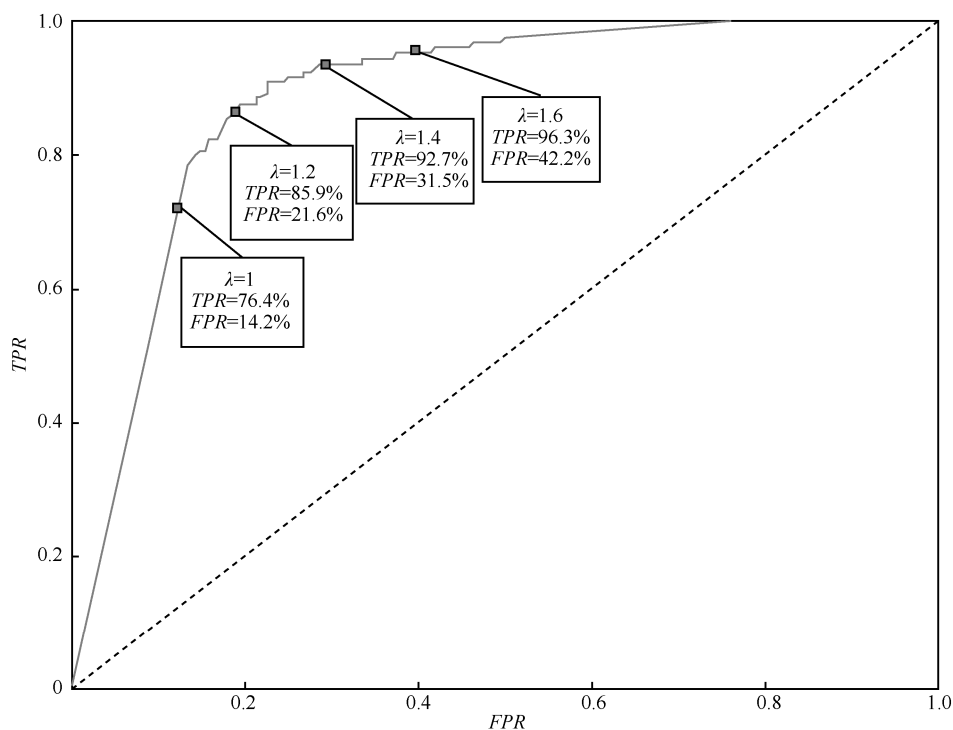


图9 不同 λ 取值下 TPR 与 FPR 的平衡曲线

KNN 算法 ($k=3, 5, 7$ 取最优)、决策树算法 C4.5、分类回归树算法 CART、和线性核函数的支持向量机 SVM 算法对 TSMWD-II 的特征提取方案所提取出的更为详尽的特征向量进行最终的分类判决, 其中训练集和测试集的选取继续选用与 TSMWD-I 相同的交叉验证的方法, 此处不再详述。将分别从检测的准确率、精准率 (precision) 和召回率 (recall) 这 3 个维度对不同分类算法的分类性能进行衡量, 其中检测的准确率用于衡量该模块做出的判决中总体的正确率情况, 精准率主要用于衡量该模块对检出的恶意样本的查准率, 召回率等同于真正率, 用于衡量模块对于恶意样本的检测能力。其实实验结果如表 2 所示, 分类结果比较直方图如图 10 所示。

表 2 TSMWD-II 不同分类算法分类结果均值比较

算法	ACC	Precision	Recall
KNN	76.31%	79.22%	78.69%
C4.5	82.15%	85.74%	85.30%
CART	86.40%	90.48%	90.44%
SVM	93.57%	92.11%	91.80%

根据该实验结果发现, 基于线性核函数的支持向量机分类算法在该模块中对恶意样本的检出率、恶意样本的查准率、整体判决结果的正确率都要优于其他 3 种分类算法, 因此可以得出结论, TSMWD 系统所使用的基于五大类的恶意网页内

容特征提取方案更加适用于 SVM 算法的分类器, 且 90% 以上的准确率基本已经达到目前静态恶意网页代码检测前沿水平, 因此选择 SVM 算法作为 TSMWD-II 模块的分类算法。

4.4 TSMWD 系统整体性能评测

在评测整个 TSMWD 系统时, 从系统判决的准确率、恶意样本检出率、正常样本的误判率这 3 个维度对系统的检测能力进行评价。同时对于系统的检测效率, 通过对测试集中的样本 (恶意样本 142 个, 正常样本 3 000 个) 进行判决的时间进行测量, 时间越短, 则检测效率越高。随着 TSMWD-I 中 λ 参数的不同, 整个系统的检测效率也会发生变化, 同 4.2 节一样, 对不同取值的 λ 进行多次测量后, 给出其变化曲线。

在两层检测系统同时使用开始前, 首先让 TSMWD-I 和 TSMWD-II 对训练集中的 3 142 个样本进行独立检测, 了解每个模块独立的检测效率。通过 10 次实验求平均后发现, TSMWD-I 对该测试集完成分类所需要的时间为 0.91 s, 而 TSMWD-II 对该测试集完成分类所需要的时间为 3.47 s, 在完全各自独立运行的情况下, TSMWD-I 的检测效率比 TSMWD-II 快了近 3.5 倍, 这主要是源于前者简易的特征提取方案, 同时易于实现的朴素贝叶斯算法也为分类判决节省了大量时间。然后对整个系统进行了检测能力和检测效率的测试, 其测试结果如表 3 所示, 比较直方图如图 11 和图 12 所示。

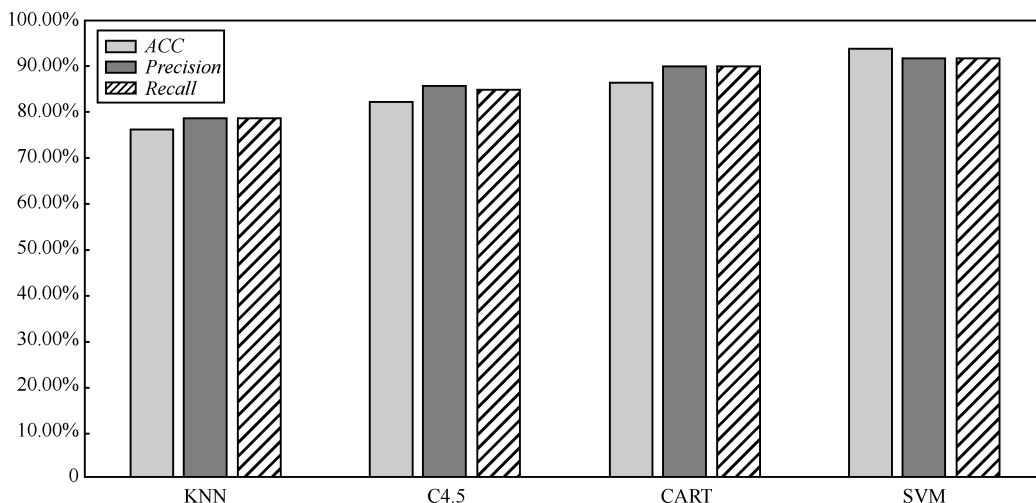


图 10 TSMWD-II 不同分类算法分类结果均值比较

表3 不同 λ 取值下的 TSMWD 综合性能

λ	检测时间/s	ACC	TPR	FPR
1	1.49	79.28%	81.10%	8.81%
1.2	1.64	86.60%	86.71%	7.95%
1.4	1.81	91.14%	91.76%	7.69%
1.6	2.24	93.14%	93.87%	7.62%
无 TSMWD-I	3.47	93.57%	94.37%	7.61%

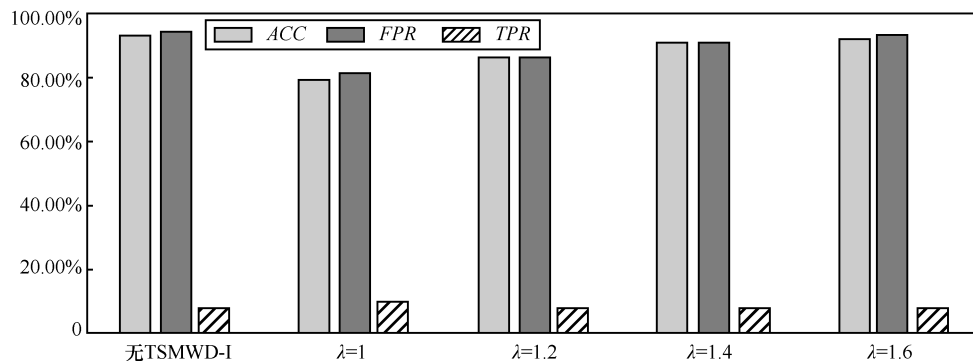
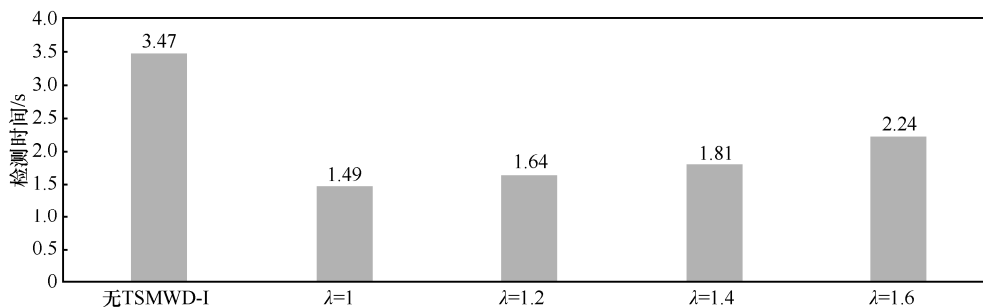
通过表3可以发现,当TSMWD-I模块不存在时,意味着此时的 λ 趋向于无穷大,即第一检测模块将所有的样本都判定为未知样本交由第二模块进行处理,此时TSMWD系统的检测效率最低,时间开销最大,为3.47 s,但是系统对于恶意样本的检测能力最强、准确率最高。当 λ 为1时,即TSMWD-I的朴素贝叶斯算法不进行任何修正,这时绝大多数的正常样本都不再经过第二检测模块进行判决,整个系统的检测效率得到显著提升,检测时间缩短为1.49 s。但是由于未经修正的第一检测模块对于恶意样本存在大量的误判,因此系统整体的检测能力受到影响,准确率下降了近15%。引入了矫正因子 λ 后,随着 λ 的不断增大,被TSMWD-I过滤的正常样本不断减

少,检测时间有所上升;但是整个系统的检测能力不断提高,检测准确率所受影响越来越小。当 λ 达到1.6时,可以发现,TSMWD系统相比于传统的只有TSMWD-II的检测系统在检测效率上由3.47 s缩短为2.24 s,检测效率提升约35%,且在系统检测能力方面由于TSMWD-I的TPR很高,所以所受的影响非常少,仅由原先的93.57%下降为93.24%,检测能力仅下降0.4%,依然维持在目前静态恶意网页检测的前沿水平。

综上所述,本文提出的基于机器学习的两层恶意网页检测系统可以在几乎不影响传统检测系统检测能力的前提下,显著提升原恶意网页检测系统的检测效率,使之更加符合当前每日新增网页数量井喷式的使用环境。

4.5 TSMWD 与现有方案的对比

经过上述实验可知,本文所提出的两层恶意网页检测系统,能够根据不同的使用环境与使用需求,通过调整参数 λ 进行检测能力和检测效率之间的相互转换。本节将本文提出的检测系统与现有的其他静态恶意网页检测系统进行比较,由于本系统在注重检测率的同时更加注重性能上的

图11 不同 λ 取值下的 TSMWD 检测能力图12 不同 λ 取值下的 TSMWD 检测效率

提升, 而很多现有的研究方案仅给出了检测能力上的评价指标, 而没有给出相应的检测速率和检测效率, 所以对比主要在两个维度上进行: 一方面与现有解决方案进行检测能力上的对比; 另一方面与给出相应检测速度的解决方案进行检测效率上的对比。

将 TSMWD 恶意网页检测系统分别与 Pawan 等^[25]提出的钓鱼网页检测系统 PhishNet、Sangho 等^[26]提出的恶意 URL 准实时检测系统 WarningBird、Likarish 等^[27]提出的恶意网页检测系统、Justin 等^[28]提出的恶意 URL 检测系统 Beyond Blacklists、Liu 等^[29]提出的钓鱼网页检测系统进行对比。

在检测效率方面, 根据文献[25]的描述, Google Safety Browser API 大约需要 80 ms 的时间为每个恶意网页 URL 在其数据库中进行查询, 这其中还需要包括查询数据分组和结果数据分组在网络中传输所消耗的时间, 而 PhishNet 将其检测速率提升了约 80 倍, 检测出每个恶意 URL 的平均消耗时间为 1 ms, 该方案在启发式规则提出的恶意 URL 检测方案的基础上进行改进, 准确率能够保证在 90% 以上。在 WarningBird 的项目中, 系统消耗的时间主要分为 3 个部分, 即域名分类、特征提取和分类模块, 主要时间消耗占比如图 13 所示, 其中对于每个 URL 检测所消耗的时间少于 3.6 ms, 由于本文的系统是没有域名分类模块的, 并且是根据网页内容对其属性进行判定, 搜集网页的数据采集模块所消耗的时间是不计算在检测系统的检测时间内的, 所以为了给出更合理的对比, 除去图 13 中耗时占比近一半的域名分组时间, 从而平均每个 URL 的

耗时大约在 1.5 ms 左右, 与 PhishNet 相差无几。对于 WarningBird 方案的准确率, 文献[26]中有相关数据, 其系统检测的准确率为 91.53%, 假正率为 1.23%。

选择 TSMWD 中 $\lambda=1.4$ 时的测试结果与 PhishNet 和 WarningBird 的检测效率及检测准确率进行横向对比, 对比结果如表 4 所示。可以看到, 在检测速度方面, TSMWD 相较于较快的 PhishNet 而言提升了约 40%, 相较于 WarningBird 提升了超过 150%; 在检测能力方面, PhishNet 提供的检测数据有限, WarningBird 检测能力与 TSMWD 相当, 准确率方面 WarningBird 略高一点点, 真正率方面相对较弱, 但总体属于同一个水平, 都已经达到了目前静态检测系统的较高水平。

表 4 TSMWD 与现有检测方案检测效率对比

检测方案	检测效率 (毫秒/个)	ACC	TPR	FPR
PhishNet	1	90%以上	—	—
WarningBird	1.5	91.53%	88.84%	1.23%
TSMWD($\lambda=1.4$)	0.58	91.14%	91.76%	7.69%

在检测能力方面, 将 TSMWD 与 Justin 等^[28]提出的 Beyond Blacklists、Likarish 等^[27]提出的混淆检测系统、Liu 等^[29]提出的自动化检测系统进行了对比, 由于这些系统在其文献中没有提及到相关的性能参数, 所以无法与之做检测效率方面的对比。TSMWD 与它们检测能力的对比如表 5 所示。Justin 的方案由于提出的时间较早, 所以检测能力上不如其余三者。其余 3 种方案的检测

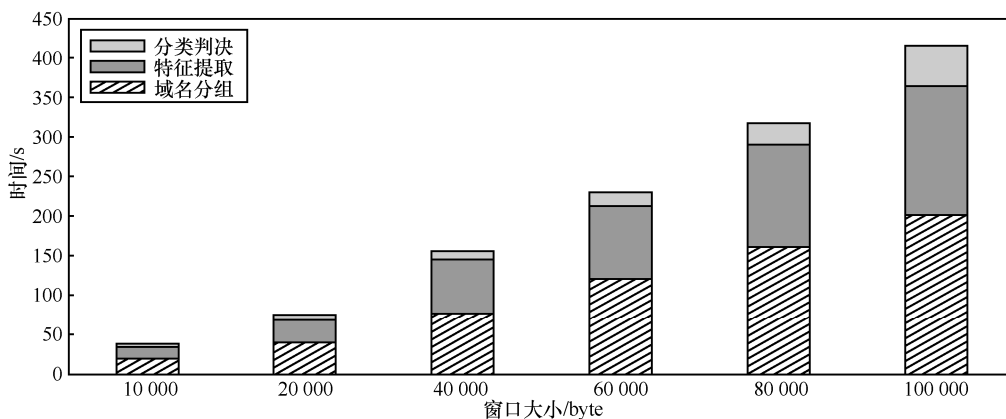


图 13 WarningBird 方案中检测时间开销

总体准确率均维持在 91%~92%，虽然说对于恶意样本和正常样本的检出率各有所长，但对于未知网页样本的总体分类能力是非常接近的。

表 5 TSMWD 与现有检测方案检测能力对比

检测方案	ACC	TPR	FPR
Justin	89.82%	91.0%	7.60%
Peter Likarish	92.0%	91.0%	7.60%
Gang Liu	91.44%	91.0%	7.60%
TSMWD($\lambda=1.4$)	91.14%	91.0%	7.60%

通过与现有静态检测方案在检测效率和检测能力指标上的对比可以发现，本文所提出的基于机器学习的两层恶意网页检测系统 TSMWD，能够在检测能力达到当今静态检测前沿水平的同时，显著提高检测的速度和效率，并且能够通过调整参数 λ 的实现检测时间和检测效率的相互转换。

5 结束语

随着互联网的不断发展和普及，网页数量正以井喷式的速度快速增加，这也为恶意网页的产生提供了温床。如何能够在最短的时间内从大量的新增网页样本中筛选出恶意样本，为恶意网页检测系统的研究提出了新的值得研究的课题。本文针对恶意网页检测中所遇到的各类问题进行了研究，针对当前传统静态恶意网页检测方案在面对海量的新增网页时面临的压力，引入了两段式的分析检测过程，并依次为每段检测提出相应的特征提取方案，通过层次化地使用优化的朴素贝叶斯算法和支持向量机算法，设计并实现了一种兼顾效率和功能的恶意网页检测系统 TSMWD。第一层检测系统用于过滤大量的正常网页，其特点为效率高、速度快、更新迭代容易，真正率优先。第二层检测系统追求性能，对于检测的准确率要求较高，时间和资源的开销上适当放宽。实验结果表明，该架构能够在整体检测准确率基本不变的情况下，提高系统的检测速度，在一定情况下，接纳更多的检测请求。

参考文献：

[1] 中国互联网信息中心. 第 37 次中国互联网络发展状况统计报告[R].

北京: CNNIC, 2016.

CNNIC. The 37th report of China Inter development statistics[R]. Beijing: CNNIC, 2016.

[2] [EB/OL].<http://www.securelist.com/en/analysis/>.

[3] PROVOS N, MAVROMMATIS P, RAJAB M A, et al. All your iFRAMES point to us[C]//Conference on Security Symposium. 2008:1-15.

[4] SHENG S, WARDMAN B, WARNER G, et al. An empirical analysis of phishing blacklists[C]//The Sixth Conference on Email and Anti-Spam (CEAS). 2009.

[5] ESHETE B, VILLAFIORITA A, WELDEMARIAM K. Malicious website detection: effectiveness and efficiency issues[C]//SysSec Workshop. 2011: 123-126.

[6] Making the Web safer[R/OL]. <http://www.google.com/transparencyreport/safebrowsing/?hl=en>.

[7] Malware domain list[EB/OL]. <http://www.malwaredomainlist.com>.

[8] OpenDNS, PhishTank[EB/OL]. <http://www.phishtank.com>.

[9] PRAKASH P, KUMAR M, KOMPELLA R R, et al. Phishnet: predictive blacklisting to detect phishing attacks[C]//INFOCOM. 2010: 1-5.

[10] CHRISTODORESCU M, JHA S. Testing malware detectors[J]. ACM Sigsoft Software Engineering Notes, 2004, 29(4):34-44.

[11] CHOU, NEIL, ROBERT LEDESMA, YUKA TERAGUCHI, et al. Client-side defense against Web-based identity theft[C]//The 11th Annual Network & Distributed System Security Symposium (NDSS). 2004:1-16.

[12] HOU Y T, CHANG Y, CHEN T, et al. Malicious Web content detection by machine learning[J]. Expert Systems with Applications, 2010, 37(1):55-60.

[13] ROESCH M. Snort-lightweight intrusion detection for networks[J]. Lisa, 1999:229-238.

[14] LIN S F, HOU Y T, CHEN C M, et al. Malicious webpage detection by semantics-aware reasoning[C]//The Eighth International Conference on Intelligent Systems Design and Applications. 2008: 115-120.

[15] ZHANG Y, HONG J I, CRANOR L F. Cantina: a content-based approach to detecting phishing web sites[C]//The 16th International Conference on World Wide Web. 2007: 639-648.

[16] HOU Y T, CHANG Y, CHEN T, et al. Malicious Web content detection by machine learning[J]. Expert Systems with Applications, 2010, 37(1):55-60.

[17] JUSTIN M, SAUL L K, SAVAGE S, et al. Beyond blacklists: learning to detect malicious Web sites from suspicious URLs[C]//The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2009: 1245-1254.

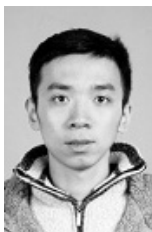
[18] YOO S, KIM S, CHOUDHARY A, et al. Two-phase malicious web page detection scheme using misuse and anomaly detection[J]. International Journal of Reliable Information and Assurance, 2014, 2(1).

[19] CANALI D, COVA M, VIGNA G, et al. Prophiler: a fast filter for the large-scale detection of malicious web pages[C]//The 20th International Conference on World Wide Web. 2011: 197-206.

[20] The German honeyclient project[EB/OL].<http://www.chicagohoneynet.org/german-honeypot-holz>.

- [21] The HoneyNet Project. Know your enemy: honeynets[EB/OL]. <http://old.honeynet.org/papers/honeynet/>.
- [22] MAYNOR D. Metasploit toolkit for penetration testing, exploit development, and vulnerability research[M]. Elsevier, 2011.
- [23] HAUTUS M L J. The formal Laplace transform for smooth linear systems[M]//Mathematical Systems Theory. Berlin:Springer, 1976: 29-47.
- [24] GOLUB G H, HEATH M, WAHBA G. Generalized cross-validation as a method for choosing a good ridge parameter[J]. Technometrics, 1979, 21(2): 215-223.
- [25] PRAKASH P, KUMAR M, KOMPPELLA R R, et al. Phishnet: predictive blacklisting to detect phishing attacks[C]//INFOCOM. 2010: 1-5.
- [26] LEE S, KIM J. Warningbird: a near real-time detection system for suspicious URLs in twitter stream[J]. IEEE Transactions on Dependable and Secure Computing, 2013, 10(3): 183-195.
- [27] LIKARISH P, JUNG E, JO I. Obfuscated malicious javascript detection using classification techniques[C]//The 4th International Conference on Malicious and Unwanted Software (MALWARE). 2009: 47-54.
- [28] MA J, SAUL L K, SAVAGE S, et al. Beyond blacklists: learning to detect malicious Web sites from suspicious URLs[C]//The 15th ACM SIGKDD international conference on knowledge discovery and data mining. 2009: 1245-1254.
- [29] LIU G, QIU B, WENYIN L. Automatic detection of phishing target from phishing webpage[C]//The 20th International Conference on Pattern Recognition (ICPR). 2010: 4153-4156.

作者简介:



王正琦(1992-), 男, 江苏镇江人, 中国科学技术大学硕士生, 主要研究方向为网络安全。



冯晓兵(1992-), 女, 山东聊城人, 中国科学技术大学硕士生, 主要研究方向为网络安全。



张驰(1977-), 男, 中国科学技术大学副教授、博士生导师, 主要研究方向为计算机网络、信息安全。