软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(2):384−397 [doi: 10.13328/j.cnki.jos.005061]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种基于主动学习的恶意代码检测方法*

毛蔚轩, 蔡忠闽, 童 力

(智能网络与网络安全教育部重点实验室(西安交通大学),陕西 西安 710049)

通讯作者: 蔡忠闽, E-mail: zmcai@mail.xjtu.edu.cn

摘 要: 现有恶意代码的检测往往依赖于对足够数量样本的分析.然而新型恶意代码大量涌现,其出现之初,样本数量有限,现有方法无法迅速检测出新型恶意代码及其变种.在数据流依赖网络中分析进程访问行为异常度与相似度,引入了恶意代码检测估计风险,并提出一种通过最小化估计风险实现主动学习的恶意代码检测方法.该方法只需要很少比例的训练样本即可实现准确的恶意代码检测,比现有方法更适用于新型恶意代码检测.通过对真实的8 340个正常进程和7 257个恶意代码进程的实验分析,与传统基于统计分类器的检测方法相比,该方法明显地提升了恶意代码检测效果.即便在训练样本仅为总体样本数量 1%的情况下,该方法也可以达到 5.55%的错误率水平,比传统方法降低了36.5%.

关键词: 访问行为;恶意代码检测;主动学习;数据流依赖网络

中图法分类号: TP309

中文引用格式: 毛蔚轩,蔡忠闽,童力.一种基于主动学习的恶意代码检测方法.软件学报,2017,28(2):384−397. http://www.jos.org.cn/1000-9825/5061.htm

英文引用格式: Mao WX, Cai ZM, Tong L. Malware detection method based on active learning. Ruan Jian Xue Bao/Journal of Software, 2017,28(2):384−397 (in Chinese). http://www.jos.org.cn/1000-9825/5061.htm

Malware Detection Method Based on Active Learning

MAO Wei-Xuan, CAI Zhong-Min, TONG Li

(Key Laboratory for Intelligent and Network Security, Ministry of Education (Xi'an Jiaotong University), Xi'an 710049, China)

Abstract: Existing techniques of malware detection depend on observations of sufficient malware samples. However, only a few samples can be obtained when a novel malware first appears in the World Wide Web, which brings challenges to detect novel malware and its variants. This paper studies the anomaly and similarity of processes with respect to their access behaviors under data flow dependency network, and defines estimated risk for malware detection. Furthermore, the study proposes a malware detection method based on active learning by minimizing the estimated risk. This method achieves encouraging performance even with small samples, and is applicable to defending against rapidly increasing novel malware. Experimental results on a real-world dataset, which consists of access behaviors of 8 340 benign and 7 257 malicious processes, demonstrate better performance of the presented method than traditional malware detection method based on statistical classifier. Even with only 1% known samples, the new method achieves 5.55% error rate, which is 36.5% lower than the error rate of traditional statistical classifier based method.

Key words: access behavior; malware detection; active learning; data flow dependency network

根据赛门铁克在全球范围监测的统计数据,2014年全年新增恶意代码3.17亿个,恶意代码总数已达17亿[1].

* 基金项目: 国家自然科学基金(61175039, 61221063, 61375040); 陕西省国际合作重点项目(2013KW11); 中央高校基本科研业务费专项资金(2012jdhz08)

Foundation item: National Natural Science Foundation of China (61175039, 61221063, 61375040); International Research Collaboration Project of Shaanxi Province (2013KW11); Fundamental Research Funds for Central Universities (2012jdhz08)

收稿时间: 2015-12-28; 修改时间: 2016-03-03; 采用时间: 2016-03-22; jos 在线出版时间: 2016-04-20

CNKI 网络优先出版: 2016-04-19 15:38:50, http://www.cnki.net/kcms/detail/11.2560.TP.20160419.1538.002.html

新型恶意代码数量的迅速增长,为恶意代码检测带来了挑战.恶意代码的检测往往依赖于对足够数量的样本及其变种的分析,并在此基础上基于人工经验或利用统计学习方法自动化地提取规则或特征,更新现有检测器[2−6].由于采集和分析足够数量的相关样本往往需要较长时间,现有检测方法无法迅速检测出新型恶意代码及其变种,新型恶意代码的大量出现仍然对整个社会构成持续性的威胁.

在新型恶意代码出现的初期,往往出现的样本数量较少,无法为目前的恶意代码检测方法提供足够信息.样本大量出现后,由于初期没有得到有效的防范,导致严重的安全事件时有发生[7].与此同时,无论是基于人工经验的方法,还是利用统计学习的方法,都是建立在充分的已知样本基础上.基于人工经验的方法通过分析新型样本的行为、指令序列,根据知识经验,创建用于标识新型恶意代码的行为规则、特征码.人工分析的方法准确率高,但耗时长,自适应性较低.随着统计学习理论的研究与应用,统计学习方法为恶意代码规则、特征的提取提供了有效的工具[2−4,6].这类方法往往依赖于充分的已知样本集合,即已经确定正常或恶意的程序样本.对于新型恶意代码样本标签的确定,往往首先需要人工手动分析以保证准确性.人工分析耗时长,很难保证短时间内得到所需的充分已知样本集合.如何在仅已知少量样本的情况下保证恶意代码检测的准确率,是应对当前新型恶意代码快速增长形势的关键问题.

为了解决这一问题,本文利用主动学习的方法,在分析系统对象的数据流依赖关系的基础上,提出了一种将程序访问行为的异常度与相似性结合的恶意代码检测方法.该方法通过增量式地训练,可以保证在少量已知样本下得到理想的准确率.在以往的研究工作中发现,通过对系统对象间访问行为的数据流依赖关系进行网络化建模,可以从网络结构的角度,分析并评价系统对象的重要性,并且有效地识别系统中安全角度下的重要对象[8].本文在系统对象重要性评价的基础上,利用程序访问对象的重要性,定义了程序访问行为的关键检测点,用以分析程序访问行为的异常度.同时,全系统下的数据流依赖关系网络为评价系统对象的相似性提供了依据,帮助我们从访问对象的角度分析进程访问行为的相似性,进而为判断程序的正常或异常标签提供了依据,即程序的访问行为越相似,其标签就越相似.结合程序访问行为的异常度与相似性,我们利用主动学习的方法实现了应对当前大量未知样本形势的有效的恶意代码检测方法.

主动学习是一种增量式的样本预测方法,在定义估计风险的基础上,从未标记样本中找出最小化估计风险的样本进行预测.在程序访问行为异常度的基础上,我们构造了统计学习分类器,用于预测未知样本的标签.同时,通过对程序访问行为相似性的分析,可以得到未知样本预测标签的估计风险.我们找出最小化估计风险的未知样本,以预测标签作为未知样本的标签,并将其作为已知样本,进一步完善基于访问行为异常度的分类器.这种最小化估计风险的策略可以不断完善,用于恶意代码检测的分类器,并在此基础上进一步降低估计风险.从而保证了即便在少量样本的情况下,也可以达到较理想的准确率,减少了人工分析样本的开销,加快了对新型恶意代码的响应,更适合于应对当前新型恶意代码快速增长的形势.通过对 8 340 个日常正常进程生命周期的访问行为以及 7 257 个真实恶意代码的访问行为的实验验证,文本的方法与原来仅使用统计分类器的方法相比,在检测效果方面呈现出了显著的增长.

本文的贡献主要包括如下 3 个部分.

- 提出了一种基于主动学习的恶意代码检测方法.该方法基于最小估计风险的策略,增量式地主动学习未知样本,不断完善恶意代码检测分类器,仅需要少量已知样本即可得到较理想的恶意代码检测效果.
- 在系统对象数据流依赖图的基础上,提出了进程访问行为异常度和相似性的评价方法.同时,从安全属性和统计特征两方面描述了访问行为,为基于主动学习的恶意代码检测提供了有效的访问行为特征.
- 真实数据集下的充分实验.通过在 8 340 个正常进程以及 7 257 个真实恶意代码的访问行为的实验,与传统基于统计分类器的方法相比,主动学习的方法提升了恶意代码检测效果.在仅已知 1%样本的情况下,检测错误率降低了 36.5%.这种基于小样本训练的检测方法将为新型恶意代码的检测提供重要的技术手段.

本文首先在第 1 节介绍恶意代码检测方面的相关工作.第 2 节概述本文提出的基于主动学习的恶意代码检测方法.第 3 节介绍系统资源的数据流依赖性网络的定义,以及在进程-文件数据流依赖网络下利用结构分析得到的对象重要性、相似性的评价.第 4 节阐述基于最小化估计风险的主动恶意代码检测方法.第 5 节描述我们的

实验数据集、实验设计及结果.最后,第 6 节总结本文的工作并讨论未来的研究方向.

1 相关工作

主动学习是一种半监督学习方法,通常可分为两种类型:(1) 用户主动标记样本;(2) 学习器主动标记样本.前一种方法在一定样本选取策略下,如不确定性、重要性等[9,10],选取部分未标记样本返回给用户,通过用户对这部分样本的人工标定,主动地从用户获取信息,提高学习器的效果.后一种方法同样根据一定的选取策略,如风险、方差最小化等[11,12],选取部分待预测样本作为已知样本,主动地从数据中获取信息并更新学习器,从而提高效果.主动学习主要用于解决样本标记开销大的问题,并应用于字体识别、计算机视觉等领域[11,12].本文在定义程序访问行为的估计风险的基础上,通过最小化估计风险的策略,从数据中主动标记未知样本,降低了对已知数据集合以及人工标定的依赖性,加快了对新型恶意代码的响应.

恶意代码检测的研究并不是一个崭新的课题.目前对恶意代码检测方法的研究可分为两种类型:(1) 静态特征分析;(2) 动态行为分析.静态特征分析主要利用可执行二进制文件格式,通过反编译的手段提取指令序列,或提取系统调用、函数调用构造控制流分析图,并以此作为特征,建立恶意代码检测模型[13,14].静态分析的方法虽然效率高,但其容易受到攻击者不断改进的混淆技术的影响.并且,越来越多的静态特征的发现,导致出现了特征泛滥的现象.动态行为分析的技术弥补了静态分析在检测效果方面的不足.本文所采用的是一种动态行为分析技术,因此我们主要对这部分相关工作进行介绍.

动态行为分析是指利用虚拟机、硬件模拟器等技术,通过 API 挂钩、污点传播等手段监控程序动态执行过程的分析方法[6,15−17].在监控得到程序动态行为的基础上,建立程序行为模型,从而进行恶意代码检测[3,5,6,16].按照分析对象的不同,我们可以将动态行为分析的方法分为两类:以程序为中心的方法(program-centric)和以系统为中心的方法(system-centric)[15].

以程序为中心的方法单独分析程序在一次执行过程中的行为,不考虑进程与操作系统的交互情况.Forrest 等人首先从系统调用序列的角度建立了基于系统调用连续子序列(*n*-gram)的程序行为异常检测模型[2].Fredrikson 等人分析了每个程序执行过程中系统调用的数据依赖关系,构造了每个程序的数据依赖行为图,基于图结构的相似性,利用图挖掘的方法进行恶意代码检测[3].Jang 等人在分析了大量程序行为特征的基础上,提出了一种基于特征哈希的行为相似性评价方法,有效地提高了恶意代码分析的效率[4].王蕊等人利用污点传播分析,构建用于描述程序行为语义的系统调用依赖特征图,在恶意代码变种的检测上取得了较好的检测效果[6].这类基于程序行为的分析方法集中在程序行为本身,忽略了程序与系统资源的访问关系以及程序行为的安全意义.随着行为特征提取方法的多样化,也带来了特征泛滥的问题[4],并且检测结果也更多地依赖于程序行为的统计特征,而不是其本身的安全属性[18].这类方法虽然可以对恶意代码变种的检测取得较好的效果,但是往往依赖于充分的已知样本集合,对于一些新型恶意代码的检测存在局限性.

以系统为中心的方法致力于分析程序在执行过程中对操作系统资源的访问,通过制定访问控制规则进行恶意代码的检测及防御.Sun 等人定义了系统资源对象的完整性,利用基于完整性的访问控制规则阻止恶意代码对系统对象的恶意访问[5],并保证软件安装过程安全性[19].Lanzi 等人基于对正常进程行为的分析,建立了系统资源对象的访问行为模型,从而进行恶意代码检测及防御[15].这些方法分析了进程行为的安全意义,提出了从操作系统角度的恶意代码防御策略.但这些策略的制定往往以充分的安全攻击、防御经验为基础,且对于不同的系统需要制定不同的策略.策略的适用范围是这些方法面临的主要问题.为了自动化地分析进程行为的安全意义,我们在以前的工作中提出了基于数据流依赖关系的系统资源重要性评价方法.该方法从系统的角度自动化地分析资源对象在安全意义上的重要性,从而为分析进程行为的安全性、异常度提供了依据[8].本文则是在此基础上,结合程序访问行为的异常度和相似性,同时从程序行为的安全属性和统计特征两个方面,提出一种有效的恶意代码检测方法.

2 方法概述

我们将恶意代码检测视为两类样本(正常、恶意)的分类问题,利用数据流依赖网络分析进程访问行为的异常度和相似性,进而提出了一种基于最小化估计风险的主动学习方法.图 1 给出了该方法的框架图.我们首先对进程访问行为进行监控,并捕获行为数据.利用正常进程的访问行为数据构造数据流依赖网络,并在该网络下分析系统对象在安全意义下的重要性以及进程访问行为的相似性.然后,根据系统对象的重要性定义每个进程访问行为中的关键检测点,并作为进程访问行为的异常度信息.利用已知标签样本的关键检测点特征,构造用于恶意代码检测的分类器.与此同时,我们假设进程访问行为越相似,其标签就越相同;如果访问行为相似,但预测标签不同,则样本预测错误的风险高.这一假设为我们定义未标记样本的估计风险提供了依据.在最小化估计风险策略下,主动学习未标记样本,增量式地将预测的部分未知样本加入已知样本集合中,进而更新检测分类器.这种方法降低了对已知样本集合规模的要求,可以有效应对当前新型恶意代码的态势,加快对其可能引发的安全事件的响应.

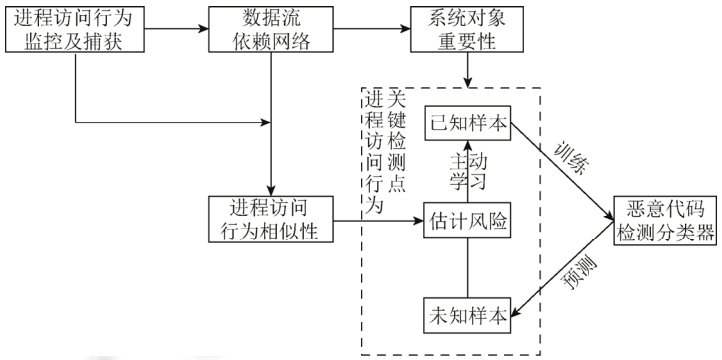


Fig.1 Overview of malware detection based on active learning

图 1 基于主动学习的恶意代码检测框架图

3 系统对象数据流依赖网络下的重要性和相似性

本文的研究主要集中在两类重要的系统对象上,进程对象和文件对象.作为操作系统中信息的承载者和传递者,进程对象和文件对象对操作系统的正常运转起到了至关重要的作用,同时也成为恶意代码攻击时的主要目标.我们将系统对象(进程、文件)访问相关的系统调用视为访问事件,进程对象在执行的生命周期中产生的访问事件序列表示进程的访问行为.系统对象的访问事件产生了系统对象间的数据流,并进一步引出了系统对象间的依赖关系.利用系统对象间的依赖关系,可以将涉及进程对象、文件对象的访问事件分为以下 3 种类型[20].

(1) 读取.如果一个进程 p 读取了文件 f 的信息,包括文件内容、文件名、修改时间等属性,则数据信息从 f 流向了 p,因此认为对象 p 依赖于对象 f.

(2) 修改.如果一个进程 p 修改了文件 f 的信息,包括写入内容、删除文件、创建文件等操作,则数据信息从 p 流向了 f,因此认为对象 f 依赖于对象 p.

(3) 创建子进程.如果一个进程 p 创建了子进程 p',则信息首先从进程 p 流向了子进程 p'的镜像文件,数据信息从子进程 p'流向了进程 p 的镜像文件,同时,信息从 p'的镜像文件流向 p',因此认为 p'的镜像文件依赖于 p,p'依赖于 p'的镜像文件,p 的镜像文件依赖于 p'.

在以上 3 种类型中,读取和修改事件比较直观,而创建子进程事件相对复杂.其具体原因我们认为是在子进程的创建过程中,通常首先由父进程将相关启动参数传递给子进程的镜像文件并启动该镜像文件,子进程通过加载其镜像文件开始执行,子进程结束后会将执行结果返回给其父进程的镜像文件,因此就引出了上述 3 个数据流的依赖关系.

以系统对象作为节点,对象间数据流的依赖关系作为有向边,可以得到系统对象数据流依赖关系网络,更确切地说,是进程-文件数据流依赖网络.该网络可表示为有向图 $G(V;E)$,其中,V 是进程、文件对象的集合,E 表示对象间依赖关系的有向边集合,我们将 v_i 依赖于 v_j 表示为存在一条有向边从 v_i 指向 v_j,即 $v_i \rightarrow v_j$ 表示系统对象 v_i 依赖于系统对象 v_j[8].

3.1 系统对象重要性评价

我们在之前的工作中提出了一种基于数据流依赖的系统对象重要性评价方法,从全系统的角度定量分析了访问对象的重要性[8].具体来说,该方法是建立在以下关于系统对象重要性的推论基础上的重要性评价方法.

- 被重要进程读取的文件是重要的文件.
- 被很多进程读取的文件可能是重要的文件.
- 修改重要文件的进程是重要的进程.
- 修改很多文件的进程可能是重要的进程.
- 创建重要进程的进程是重要的进程.
- 创建很多进程的进程可能是重要的进程.

我们使用在网页重要性排序中较为成熟的 PageRank 方法作为评价系统对象重要性的指标.令 A 表示系统对象依赖网络的邻接矩阵,则系统对象的重要性向量 p 可用如下公式计算得到:

$$\boldsymbol{p}=\frac{(1-c)}{n}\vec{1}+c\left[\left[\frac{1}{d_1},...,\frac{1}{d_n}\right]A+D\right]\boldsymbol{p},\quad D=\frac{1}{n}\left[1_A(d_1=0),...,1_A(d_n=0)\right]^T\vec{1}^T,\quad p_i=\frac{(1-c)}{n}+c\left(\sum_{j:d_j\neq 0}p_j\frac{A_{ji}}{d_j}+\sum_{j:d_j=0}\frac{p_j}{n}\right),$$

其中,c 是衰减因子表示随机游走过程的重新启动.d_j 表示对象 j 的出度.$1_A(x)$表示指示函数,当 x 的判断结果正确时函数取值为 1,否则取值为 0.利用 PageRank 对系统对象重要性进行量化评价,为进程行为检测点的选取提供了有效依据,为恶意代码检测提供了基础.

3.2 系统对象相似性评价

本文假设访问行为相似的进程,其正常、恶意标签相似.因此,某一进程行为与其他进程行为的相似性隐含了该进程的标签信息.在我们的进程-文件数据流依赖网络下,进程访问行为的相似性可以通过其在网络结构下的相似性进行衡量.衡量网络结构中两个节点之间的相似性,为揭示网络结构、预测网络演变、个性化推荐等提供了有力的依据.同时,我们的数据流依赖网络结构从全系统的角度刻画了系统对象间的关系,在此网络结构下衡量节点的相似性,为从全系统的角度衡量进程行为的相似性提供了依据.

基于网络结构的相似性评价往往可以从待评价节点的邻居节点集合进行.对于一个节点 v_i,令 $\Gamma(v_i)$ 表示节点 v_i 在网络结构上的邻居节点.基于邻居节点的相似性评价的直观解释是,两个节点 v_i,v_j 的邻居节点集合 $\Gamma(v_i)$, $\Gamma(v_j)$ 越相似,v_i 和 v_j 就越相似.下面分别给出本文使用的基于邻居节点的相似性评价方法.

3.2.1 公共邻居(common neighbor)

公共邻居这一指标利用网络结构下两个节点的公共邻居的数量评价这两个节点的相似性[21].具体来说,令 $\Gamma(v_i)$ 表示网络结构下节点 v_i 的邻居节点,则基于公共邻居的两个节点 v_i,v_j 相似性定义为 $Sim(v_i,v_j)=|\Gamma(v_i)\cap\Gamma(v_j)|$,其中,$|\cdot|$表示集合"·"中元素的数量.考虑到进程-文件数据流依赖网络是一个有向图网络,我们定义在该网络下,节点 v_i,v_j 的基于公共邻居的相似性为 $Sim(v_i,v_j)=\left|\Gamma_{\text{in}}(v_i)\cap\Gamma_{\text{in}}(v_j)\right|+\left|\Gamma_{\text{out}}(v_i)\cap\Gamma_{\text{out}}(v_j)\right|$. 其中,$\Gamma_{\text{in}}(v_i)$ 表示节点 v_j 的入邻居集合,即存在有向边指向 v_i 的邻居节点集合;$\Gamma_{\text{out}}(v_i)$ 表示节点 v_i 的出邻居集合,即存在有向边从 v_i 指向的邻居节点集合.

3.2.2 Jaccard 系数

Jaccard 系数是信息检索领域中常用的相似度衡量指标,用来衡量 v_i,v_j 两个节点都具有某一邻居的概率[15].在进程-文件数据流依赖网络中,基于 Jaccard 系数的相似性定义为

$$Sim(v_i,v_j)=\left|\Gamma_{\text{in}}(v_i)\cap\Gamma_{\text{in}}(v_j)\right|\Big/\left|\Gamma_{\text{in}}(v_i)\cup\Gamma_{\text{in}}(v_j)\right|+\left|\Gamma_{\text{out}}(v_i)\cap\Gamma_{\text{out}}(v_j)\right|\Big/\left|\Gamma_{\text{out}}(v_i)\cup\Gamma_{\text{out}}(v_j)\right|.$$

3.2.3 Adamic/Adar

Adamic 和 Adar 于 2003 年提出了一种考虑公共特征权重的衡量方法[21].与以往基于公共邻居的评价方法相比,该方法考虑了公共邻居本身的独特性,即广泛存在的公共邻居对节点间相似性计算贡献较少,而分布较少的公共邻居对于节点间相似性的衡量更有价值.在进程-文件数据流依赖网络中,基于 Adamic/Adar 的相似性定

义为

$$Sim(v_i,v_j)=\sum_{v_k\in\Gamma_{\text{in}}(v_i)\cap\Gamma_{\text{in}}(v_j)}\frac{1}{\log|\Gamma_{\text{in}}(v_k)|}+\sum_{vk\in\Gamma_{\text{out}}(v_i)\cap\Gamma_{\text{out}}(v_j)}\frac{1}{\log|\Gamma_{\text{out}}(v_k)|}.$$

以上 3 种基于邻居节点的相似性评价方法从不同的角度衡量了网络结构下节点的相似性,为我们从全系统的角度全面地衡量进程访问行为的相似性、分析进程访问行为提供了依据.我们在本文的实验中通过对比,从多个角度分析这 3 种相似性评价方法的特点.

4 基于最小化估计风险的主动恶意代码检测

4.1 基于访问行为异常度的恶意代码检测

绝大多数恶意代码通过对重要对象的敏感操作,达到其感染或破坏操作系统的目的.这里,我们将对象的修改及进程的创建操作定义为敏感操作,以进程执行敏感操作的对象为特征.根据对象的重要性建立进程访问行为的特征向量,然后利用统计学习分类器可以对正常进程和恶意代码进行分类,从而解决恶意代码检测的问题.首先,我们定义了用于进程访问行为异常检测的关键检测点:进程执行敏感操作的所有对象中最重要的 C 个对象(重要性值最高的 C 个对象).我们为每个进程构造一个长度为 C 的特征向量 x,其中,$x_i(1\leqslant i\leqslant C)$表示在该进程所有执行敏感操作的对象中,第 i 个重要对象的重要性.该特征向量描述了该进程的关键检测点信息.我们在实验中发现,在 C=5 的情况下,可以从检测率及误报率两方面得到较好的恶意代码检测效果.因此,本文的实验均在 C=5 的情况下进行.

4.2 基于访问行为相似性的估计风险

进程行为越相似,其标签就越相同,进程行为的相似性为恶意代码检测提供了估计风险的依据.估计风险的定义建立在数据样本一致性假设的基础上:(1) 相似的样本具有相同的标签;(2) 在相同局部结构(簇、流形)下的样本具有相同的标签[10,12].利用访问行为异常度构造的检测分类器 h 可以预测每个进程样本 i 的标签 $h(i)$,也称为 i 的估计标签.在我们的问题中,$h(i)\in\{0,1\}$.在访问行为相似性的假设下,我们定义分类器 h 的估计风险为

$$L(h)=\frac{1}{2}\sum_{i,j}\frac{Sim(v_i,v_j)}{\sum_k Sim(v_i,v_k)}(h(i)-h(j))^2 \tag{1}$$

上式的直观解释为,$(h(i)-h(j))^2$ 表示两个样本标签的差异,$Sim(v_i,v_j)$表示两个样本的相似性.当两个样本的相似性较高,但标签差异较大时,会导致较大的估计风险.当样本标签相似,或其本身相似性较低时,不会引起估计风险的增加.这里,我们使用第 3.2 节中的相似性评价方法作为两个进程样本 i,j 的相似性,即 $Sim(v_i,v_j)$.在此估计风险的定义下,我们基于最小化估计风险的策略,提出了一种基于主动学习的恶意代码检测方法.

4.3 基于最小化估计风险的主动学习

主动学习是机器学习中的一个重要领域,研究增量式的样本标记方法.基于最小化估计风险的主动学习方法最早由 Zhu 等人提出来[12],给定部分已标记的样本,通过主动学习从未标记样本中找出可标记样本.与随机选取不同,这种方法选择性地选取未标记样本作为新一轮学习中的已标记样本,因此也称为一种半监督学习.

我们在最小化估计风险估计的策略下,利用贪婪选择对未标记样本进行主动学习.根据已知样本的关键检测点训练分类器,预测未知样本.同时,根据未标记样本的相似性,衡量每个未标记样本产生的估计风险.选择估计风险最小的未标记样本作为标记样本,对分类器进行再学习,重复此过程,直至完成所有样本的标记.如果相似值定义合理,可以帮助学习器完善学习过程,达到更好的分类效果.根据式(1)定义的估计风险,基于最小化估计风险的主动学习是一个贪婪地寻找满足如下条件的未知样本的过程:

$$\arg\min_{i\in U}\hat{R}_i,\ \hat{R}_i=\frac{1}{2}\sum_{j\in L}\frac{Sim(v_i,v_j)}{\sum_k Sim(v_i,v_k)}(h(i)-h(j))^2 \tag{2}$$

其中,L(labeled)表示已标记样本集合,即已知正常、恶意标签的进程;U(unlabeled)表示未标记样本集合.算法 1 描

述了本文中基于最小化估计风险的主动恶意代码检测的过程.其中,在步骤 4 中,我们使用统计学习分类器得到基于进程访问行为异常度的恶意代码检测模型(分类器 h).然后在最小化估计风险的策略下,利用访问行为相似性寻找估计风险最小的未标记进程,按照 h 的预测结果对选取的进程进行标记.将带有预测标签的进程样本加入训练集 L 中,重新训练基于异常度的恶意代码检测模型,直到所有未标记的进程样本均完成标记为止.

算法 1. 基于最小化估计风险的主动恶意代码检测.

输入:初始已标记样本 L^0,未标记样本 U,每轮主动学习样本数 K.

输出:U 中样本的标签.

1: $L=L^0$,$h(j)=j$ 的真实标签, $\forall j \in L^0$
2: 计算 U 中的每个样本 i 与 L 中每个样本 j 的相似性,$Sim(v_i,v_j)$
3: **while** U 中存在未标记样本 **do**
4: 利用统计分类器在 L 上训练,得到分类器 h
5: 利用分类器 h 预测 U 中样本标签
6: 根据式(2)计算 U 中每个样本的估计风险 $\hat{R}_i$
7: **for** i=0→K−1 **do**
8: 根据式(2)选取 $k\leftarrow \arg\min_i \hat{R}_i$
9: $L \leftarrow L \bigcup \{k\}$
10: $U \leftarrow U/\{k\}$
11: **end for**
12: **end while**

在主动学习过程的步骤 6 中,算法复杂度为 $O(|U||L|)$,也即 $O(n^2)$.为了降低算法复杂度,我们对相似性的计算进行修剪.本文中我们使用 Top-N 修剪:通过设定 N,对于 U 中的样本 i,选取 L 中与 i 相似性最大的 N 个样本,即构造未标记样本的局部流形结构.这种方法最早出现在 Roweis 等人的 LLE 方法中[22].基于以往的工作[22]以及我们在实验中的观察,当 N=3 时,检测效果提升最为显著.本文所描述的实验结果均是在 N=3 的情况下得到的.

利用主动学习的方法可以有效地应对已知正常、恶意进程数量有限的情况,通过最小化估计风险的策略完成模型的建立.这种方法可以有效地应对当前互联网环境下恶意代码迅猛增长的态势.

5 实验分析

5.1 数据集描述

本文实验主要关注 Windows 平台下 32 位可执行文件.我们从 VXHeaven 下载到恶意代码样本,并从中随机挑选了 7 257 个恶意代码样本.利用 VMWare 虚拟机以及 Process Monitor 实现了沙盒系统,用以记录恶意代码对文件、进程的访问操作,目标系统为 Windows XP sp3.同时,我们在 1 台目标系统同样为 Windows XP sp3 的正常用户的机器上部署 Process Monitor,记录正常程序的文件、进程访问行为,采集过程持续了 14 天,共得到来自 155 个日常正常程序的 8 340 个进程完整生命周期内的文件和进程访问行为记录.

5.2 实验设计

本文的实验主要从 3 个方面进行:(1) 不同相似性评价方法($Sim(v_i,v_j)$)对检测结果的影响;(2) 不同分类器(h)对检测结果的影响;(3) 不同初始训练集尺寸(L^0)对检测结果的影响.我们首先针对第 3.2 节中提出的 3 种不同的网络结构下节点相似性的评价方法进行实验,对比不同评价方法对检测结果的影响.在这个实验中,我们使用最近邻居作为统计学习分类器,进行恶意代码检测.接着,我们分别使用两种不同的分类器,最近邻居(k-nearest neighbour,简称 knn)和随机森林(random forest),分析分类器对检测结果的影响.初始训练集尺寸往往是影响分类器效果的主要因素,为了应对当前数量快速增长的恶意代码,我们需要即使在较小的初始训练集下也可以达到较好检测结果的方法.因此,我们最后分析了不同初始训练集下的检测结果,以说明我们方法的实用性.我们在所

有实验中,均在指定初始训练集尺寸下进行了 10 次实验,以保证实验的随机性.

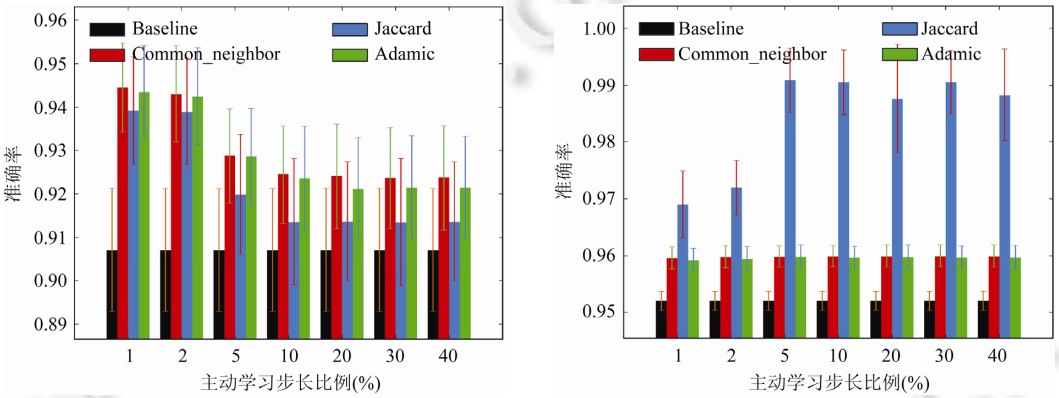
在评价检测效果方面,我们使用误报率(false positive rate)和检测率(true positive rate)来评价检测结果,使用准确率(accuracy)和错误率(error)对比不同方法的检测效果.具体来说,假设在被检测方法判断为恶意的样本中,实际为正常样本的数量为 FP,实际为恶意样本的数量为 TP;在被检测方法判断为正常的样本中,实际为正常样本的数量为 TN,实际为恶意样本的数量为 FN.误报率、检测率、准确率、错误率的定义如下:

$$\text{误报率}=\frac{FP}{TN+FP},\ \text{检测率}=\frac{TP}{TP+FN},\ \text{准确率}=\frac{TP+TN}{TP+FN+TN+FP},\ \text{错误率}=\frac{FP+FN}{TP+FN+TN+FP}.$$

5.3 实验结果

5.3.1 不同相似性评价方法对结果的影响

我们通过实验对比第 3.2 节提出的 3 种不同网络结构下节点的相似性评价对恶意代码检测结果的影响.如图 2 所示,我们选取了两种大小的初始训练集 $\left|L^0\right|\Big/\left(\left|L^0\right|+\left|U\right|\right)$,分别比较了在最近邻居分类器下,不使用(baseline,仅在初始训练集上训练分类器)和使用主动学习的方法下的检测结果.同时,我们也比较了不同比例的主动学习样本步长对检测结果的影响,即在算法 1 的输入中,$K=\left(\left|L^0\right|+\left|U\right|\right)\times$步长比例.由于每种实验设定下进行了 10 次实验,图 2 显示了 10 次实验的平均准确率及其标准差.



(a) 初始训练集大小为 1%　　(b) 初始训练集大小为 50%

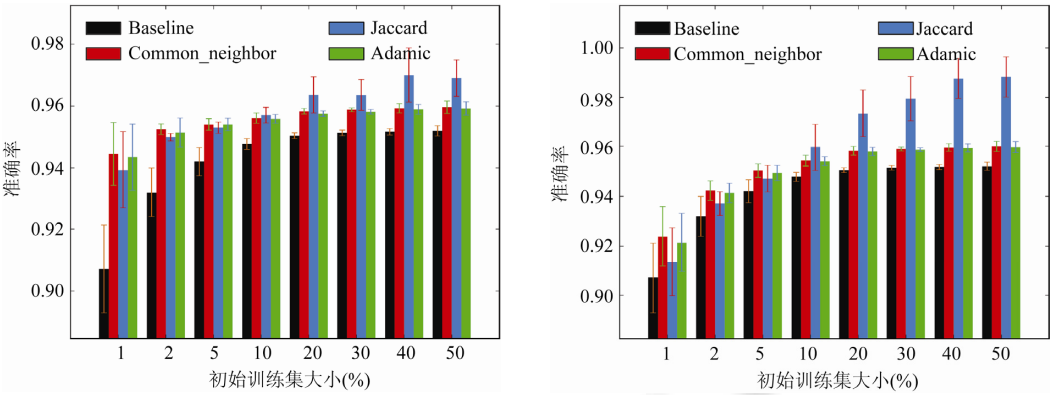
Fig.2 Accuracy of malware detection with different similarity metrics and different learning rates

图 2 不同相似性评价及不同主动学习步长下恶意代码检测的准确率

从图 2(a)中可以看出,在仅选取 1%样本作为初始训练集的情况下,主动学习的方法随着主动学习样本比例的增加而下降.当每轮主动学习 1%或 2%的样本时,baseline 的方法效果小于主动学习方法的检测准确率的 95%的置信区间,这说明在两种情况下,3 种主动学习的方法均显著地优于不使用主动学习的方法,并且,使用公共邻居的相似性评价方法优于其他两种评价方法.另一方面,从图 2(b)中可以看出,在选取 50%样本作为初始训练集的情况下,在所有主动学习样本比例下,3 种主动学习的方法均显著地优于不使用主动学习的方法,并且,与图 2(a)相反,Jaccard 系数的相似性评价方法一致地优于其他两种评价方法.与此同时,随着主动学习样本比例的增加,基于公共邻居和 Adamic 的主动学习方法的准确率并没有显著变化.基于图 2 的观察我们推测,不同相似性评价方法在不同训练集大小下,对恶意代码检测的提升效果不同.为了验证此推测,图 3 显示了在两种主动学习步长比例下(1%和 50%),3 种相似性评价得到的主动学习方法在不同初始训练集下的检测准确率.

对比图 3(a)和图 3(b),在两种主动学习步长比例下,主动学习方法在准确率上的提升随初始训练集的变化趋势相同,并且,Jaccard 系数在已知较大训练样本的情况下,可以得到优于公共邻居和 Adamic 的检测准确率.在已知较小初始训练集下,基于公共邻居的主动学习方法可以得到更好的检测准确率.更进一步地,恶意代码检测往往从误报率和检测率两个方面关注方法的检测效果.图 4 显示了 3 种基于相似性的主动学习方法在不同初始训

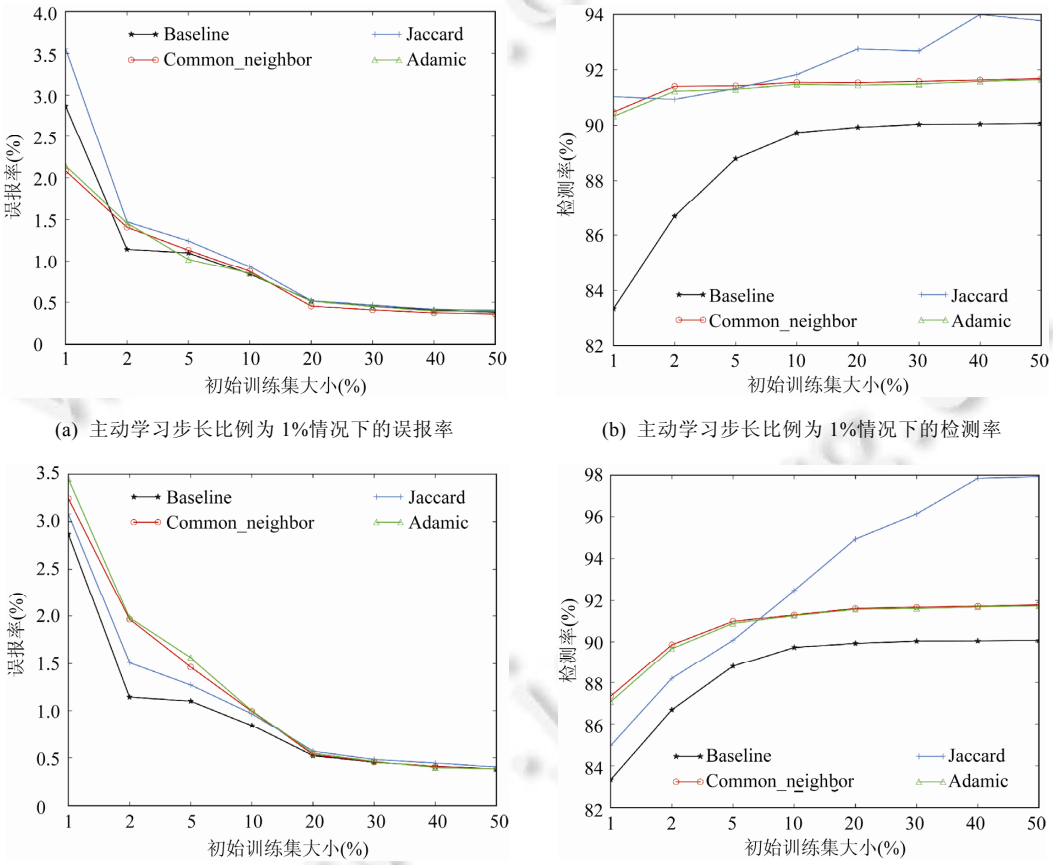
练集下的平均误报率和检测率.



(a) 主动学习步长比例为 1% (b) 主动学习步长比例为 40%

Fig.3 Accuracy of malware detection with different initial training ratios

图 3 不同初始训练集下恶意代码检测的准确率



(a) 主动学习步长比例为 1%情况下的误报率 (b) 主动学习步长比例为 1%情况下的检测率

(c) 主动学习步长比例为 40%情况下的误报率 (d) 主动学习步长比例为 40%情况下的检测率

Fig.4 False positive rate and true positive rate of malware detection with different initial training ratios

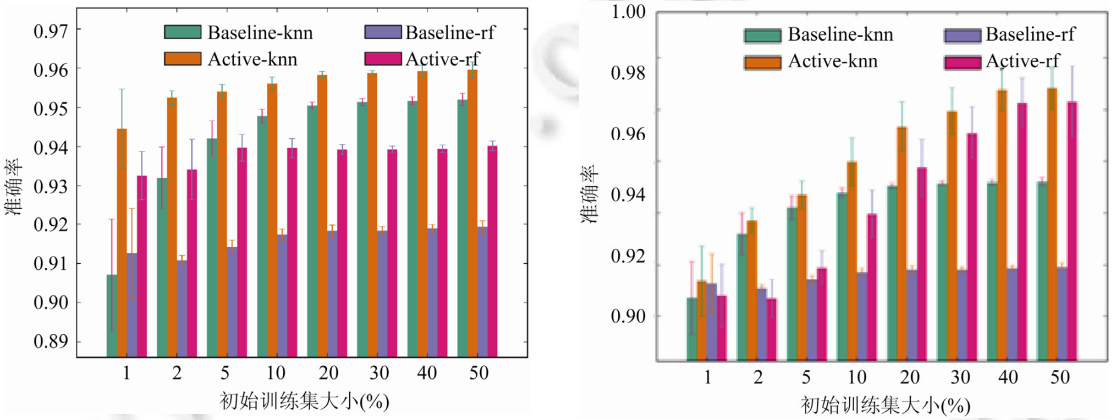
图 4 不同初始训练集下恶意代码检测的误报率和检测率

从图 4 可以看出,主动学习的方法在大于等于 10%的初始训练集下得到的误报率与不使用主动学习的方法

相似.其中,Jaccard 系数在小于 10%初始训练集下会引入更多的误报,基于公共邻居和 Adamic 的主动学习方法误报率相似.在检测率方面,3 种相似性评价下的主动学习方法均有明显的提升,且 Jaccard 系数在大于 10%初始训练集下有最大的提升.综合以上分析,我们认为加入主动学习可以明显地提升恶意代码的检测效果,在仅已知少量训练数据(<10%)或每轮主动学习较小数量的样本的情况下,基于公共邻居或 Admaic 的主动学习方法可以更大地提升检测效果.在已知大量训练数据或每轮主动学习较大数量的样本的情况下,基于 Jaccard 系数的主动学习的检测效果更优.

5.3.2 分类器对结果的影响

本文提出的基于进程访问行为异常度及相似性的恶意代码检测方法是在统计分类器基础上的主动学习的方法,统计分类器的选取也是影响检测效果的因素.因此,我们选取了常用的两种分类器:最近邻居和随机森林,分析了不同分类器下主动学习方法的检测效果.图 5 显示了两种分类器在不同初始训练集下不使用(baseline)和使用(active)主动学习的方法的检测准确率.由上一节的实验结果可知,在较小/较大数量主动学习样本的情况下,基于公共邻居/Jaccard 系数的方法检测效果最好,因此,我们分别基于公共邻居主动学习步长比例为 1%(如图 5(a)所示)和基于 Jaccard 系数主动学习步长比例为 40%(如图 5(b)所示)来对比两种分类器的检测效果.



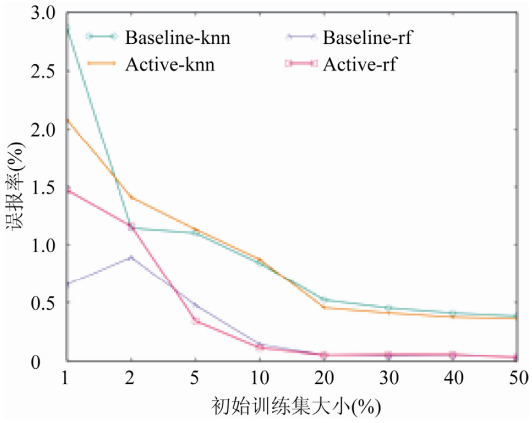
(a) 基于公共邻居主动学习步长比例为 1%　　(b) 基于 Jaccard 系数主动学习步长比例为 40%

Fig.5 Accuracy of malware detection based on knn and random forrests with different initial training ratios

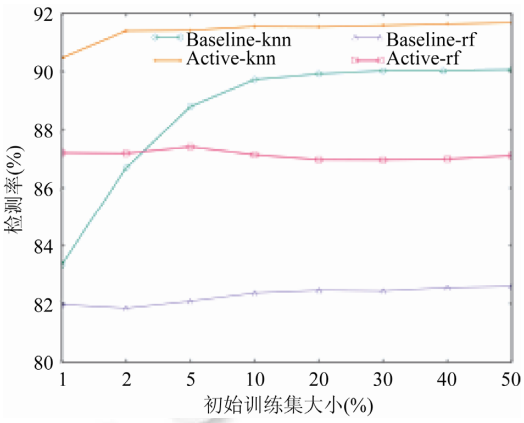
图 5 基于最近邻居和随机森林分类器的恶意代码检测在不同初始训练集下的准确率

从图 5(a)中可以看出,在较小数量的主动学习样本下,主动学习显著地提升了分类器的检测准确率,并且随机森林分类器的提升幅度更大.但在此情况下,随着初始训练集的增加,随机森林的检测效果并没有明显的增长.而在较大数量的主动学习样本下,如图 5(b)所示,随着初始训练集的增加,加入主动学习后的分类器的检测效果均有明显的增长.但在此情况下,当已知较小初始训练集(1%,2%)时,较大样本数量的主动学习反而降低了随机森林的检测效果.其原因可能是,当初始训练样本数量较小时,并不能为大量样本的主动学习提供足够的依据,主动学习的方法同时受到初始训练集大小和每轮主动学习样本数量的影响.我们在下一节的实验中将进一步分析这两个因素.同时,从图 5 中还可以看出,引入主动学习后,最近邻居分类器的检测准确率总是高于随机森林分类器的准确率.为了进一步分析,图 6 从误报率和检测率两个方面显示了两种分类器的检测效果.

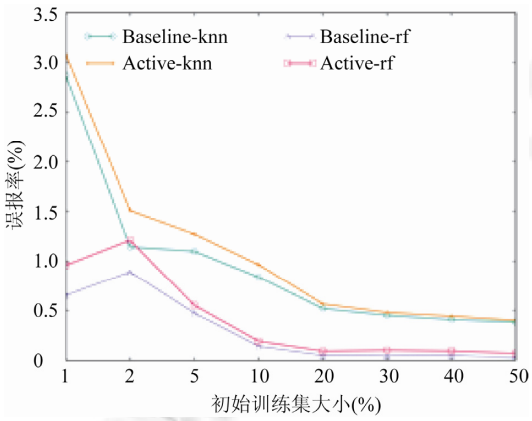
从图 6(a)和 6(c)中可以发现,在初始训练集小于 10%的情况下,加入主动学习均会提高两种分类器的误报率.并且,使用随机森林分类器的误报率明显地低于最近邻居分类器的误报率.在检测率方面,如图 6(b)和图 6(d)所示,在所有初始训练集下加入主动学习均会提高检测率,且最近邻居的检测率高于随机森林的检测率.因此我们建议,在个人、企业、政府等机构面临的真实恶意代码环境中,应首先分析对不同错误率的容忍情况,再选择适应的分类器.如,个人用户要求高可用性,可以牺牲检测率来降低误报;重要职能机构对检测率要求高,可容许部分误报.



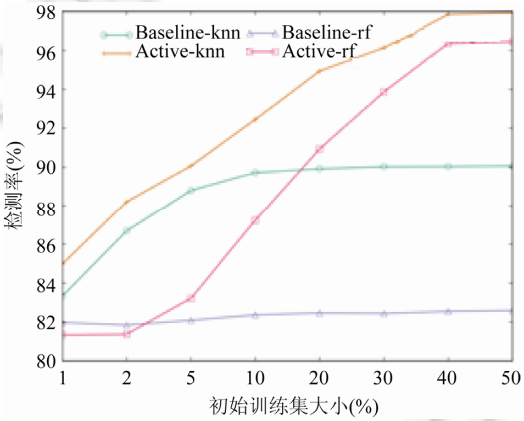
(a) 公共邻居主动学习步长比例为 1%的误报率



(b) 公共邻居主动学习步长比例为 1%的检测率



(c) Jaccard 系数主动学习步长比例为 40%的误报率



(d) Jaccard 系数主动学习步长比例为 40%的检测率

Fig.6 False positive rate and true postive rate of malware detection based on knn and random forrests with different initial training ratios

图 6 基于最近邻居和随机森林分类器的恶意代码检测在不同初始训练集下的误报率和检测率

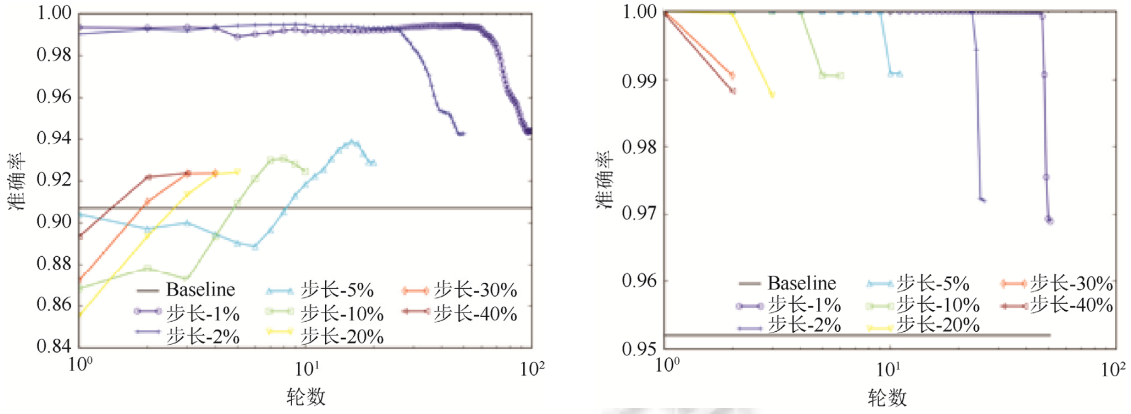
5.3.3 初始训练集尺寸与主动学习样本比例对结果的影响

从上一节的实验中我们发现,在不同大小的初始训练集下,相同的主动学习样本数量对检测结果产生不同的影响.并且,每轮主动学习样本数量的选取也会直接影响完成所有样本预测所需要的轮数,从而影响恶意代码检测效率.为了进一步理解每轮主动学习对检测结果的影响,图 7 显示了最近邻居分类器在不同初始训练集下,每轮主动学习后标定的初始测试样本的准确率.

从图 7(a)中可以看出,在已知较少训练集(1%)的情况下,每轮少量样本的主动学习(1%,2%)在前面几轮主动学习过程中可以有效地保证准确率.这说明了最小化估计风险可以为恶意代码检测提供有效的样本主动学习策略,从而提高恶意代码检测的准确率.但进程样本中存在部分当前分类器不可分的访问行为,目前基于最小化估计风险的主动学习方法并不能对这部分样本进行有效的区分.这需要我们提出更有效的分类方法,进一步提高检测效果.另一方面,在已知训练集充足的情况下(50%),如图 7(b)所示,前几轮的主动学习均能有效地提高检测准确率.但当主动学习样本数量较少(1%,2%)时,在最后几轮学习中,准确率同样出现了明显的下降.

我们进一步分析了在 1%和 50%的初始训练集下,步长比例为 1%的主动学习过程.我们发现,在最后几轮的主动学习过程中,根据式(1)得到的未知样本的估计风险出现了明显的上升.并且,存在一些较高估计风险的样本与很多未标记样本的访问行为相似,在这些样本上的错误预测,直接导致了后续在与其相似的其他未标记样本

上的错误.总体来说,在主动学习方法的基础上,这部分高风险进程数量较小,在实际恶意代码环境中,可以通过人工分析估计风险较高的样本配合主动学习更有效地完成恶意代码检测.



(a) 公共邻居初始训练集为 1% (b) Jaccard 系数初始训练集为 50%

Fig.7 Accuracy of active learning with nearest neighbor classifier in each round

图 7 最近邻居分类器每轮主动学习后的准确率

为了给不同真实环境下恶意代码检测方法的选取提供依据,我们分析了不同大小的初始训练集下检测效果最好的方法.表 1 给出了不同实验设定(初始训练集尺寸、主动学习步长比例)下,平均错误率最低的方法(分类器、相似性评价方法).

Table 1 Classifier with the lowest average error rate and its average error rate with different initial training ratios

表 1 不同初始训练集比例下平均错误率最低的分类器及其平均错误率

主动学习步长比例	初始训练集					
	1%	2%	5%	10%	20%	50%
1%	**knn-cn, 5.55%**	**knn-cn, 4.75%**	knn-ada, 4.59%	knn-jac, 4.30%	knn-jac, 3.64%	knn-jac, 3.10%
2%	knn-cn, 5.70%	**knn-cn, 4.75%**	knn-cn, 4.56%	knn-jac, 4.19%	knn-jac, 3.36%	knn-jac, 2.80%
5%	rf-cn, 7.06%	knn-cn, 4.95%	**knn-cn, 4.53%**	knn-jac, 3.95%	knn-jac, 2.92%	**knn-jac, 0.91%**
10%	rf-cn, 7.16%	knn-cn, 5.54%	knn-ada, 4.66%	**knn-jac, 3.78%**	knn-jac, 2.90%	knn-jac, 0.94%
20%	rf-cn, 7.10%	knn-cn, 5.62%	knn-cn, 4.74%	knn-jac, 3.90%	knn-jac, 2.78%	knn-jac, 1.24%
30%	rf-cn, 7.16%	knn-cn, 5.68%	knn-cn, 4.95%	knn-jac, 3.96%	knn-jac, 2.67%	knn-jac, 0.94%
40%	rf-ada, 7.03%	knn-cn, 5.77%	knn-cn, 4.97%	knn-jac, 4.02%	**knn-jac, 2.65%**	knn-jac, 1.17%
最低错误率 Baseline	rf, 8.74%	knn, 6.80%	knn, 5.80%	knn, 5.22%	knn, 4.96%	knn, 4.80%
错误率降低比例	36.50%	30.15%	21.90%	27.59%	46.57%	81.04%

cn: 公共邻居 ada: Adamic jac: Jaccard 系数

表 1 中第 2 行~第 8 行的每个单元显示了在所在列表示的初始训练集尺寸,所在行表示的主动学习步长比例下,平均检测错误率最低的检测方法(分类器-相似性),以及该方法的平均错误率.第 9 行显示了不加入主动学习(baseline)时,两种分类器(最近邻居、随机森林)中错误率较低的分类器及其检测错误率.并且,我们强调了在每一列中平均错误率最低的主动学习方法.如,当仅已知 1%的训练样本时,使用公共邻居作为相似性评价,每轮主动学习 1%样本的最近邻居分类器得到最低的平均检测错误率,5.55%.最后一行则显示了相对于仅利用分类器的方法(baseline),平均错误率最低的主动学习方法可以降低错误率的比例.可以看出,当已知训练集尺寸增加时,提高主动学习步长比例可以得到更好的检测结果.与第 5.3.1 节观察相同,当已知训练集样本较少(≤5%)时,基于公共邻居的相似性评价可以得到更好的检测效果;当已知训练集样本较大(≥10%)时,基于 Jaccard 系数的相似性评价可以得到更好的检测效果.与此同时,我们发现,当选取主动学习步长比例与初始训练集比例相近时,无论选取何种分类器和相似性评价方法,主动学习方法均显著地降低了检测错误率.在我们的实验中,主动学习的恶意代码检测方法与仅利用分类器的检测方法相比,在错误率方面平均降低了 40.6%;在仅已知 1%样本的情

况下,错误率降低了 36.5%.

6 总结与未来工作

在新型恶意代码出现之初,可供分析的样本数量有限,为及时、有效地检测新型恶意代码及其变种带来了挑战.本文从操作系统中资源访问的数据依赖网络出发,提出了一种基于主动学习,结合进程访问行为异常度及相似性的恶意代码检测方法.我们分析了资源访问行为的数据依赖关系,构造了用于描述全系统资源访问的数据依赖网络.并且,通过分析网络结构,找到进程访问的重要资源对象,并设计了基于进程访问行为异常度的恶意代码检测方法.同时,在网络结构下衡量进程访问行为的相似性,利用主动学习的方法增量式地学习用于恶意代码检测的分类器,从而提出一种将进程访问行为异常度与相似性结合的恶意代码检测方法.通过对 Windows XP sp3 用户日常使用的 8 340 个正常进程以及 7 257 个恶意代码样本的实验验证,我们发现,与传统基于统计分类器的方法相比,主动学习的方法提升了恶意代码检测效果.并且在仅已知 1%样本的情况下,本文方法与传统基于统计分类器的检测方法相比,在错误率方面降低了 36.5%,这说明我们的方法能够有效地应对当前新型恶意代码快速增长的形势.

本文基于固定步长的主动学习方法进行恶意代码检测.我们在实验中发现,主动学习步长的选取在一定程度上影响恶意代码检测效果.如何自适应地调整步长将在后续的研究工作中展开.与此同时,正如实验结果所示,在主动学习下,仍然存在少数难以检测的恶意代码,与人工分析相结合可以进一步提升检测效果.

References:

[1] Symantec. Internet security threat report. Vol.20, 2015. https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932_GA-internet-security-threat-report-volume-20-2015-social_v2.pdf

[2] Forrest S, Hofmeyr SA, Somayaji A, Longstaff TA. A sense of self for UNIX processes. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE, 1996. 120−128. [doi: 10.1109/SECPRI.1996.502675]

[3] Fredrikson M, Jha S, Christodorescu M, Sailer R, Yan XF. Synthesizing near-optimal malware specifications from suspicious behaviors. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE, 2010. 45−60. [doi: 10.1109/SP.2010.11]

[4] Jang J, Brumley D, Venkataraman S. Bitshred: Feature hashing malware for scalable triage and semantic analysis. In: Proc. of the 18th ACM Conf. on Computer and Communications Security. New York: ACM, 2011. 309−320. [doi: 10.1145/2046707.2046742]

[5] Sun WQ, Sekar R, Poothia G, Karandikar T. Practical proactive integrity preservation: A basis for malware defense. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE, 2008. 248−262. [doi: 10.1109/SP.2008.35]

[6] Wang R, Feng DG, Yang Y, Su PR. Semantics-Based malware behavior signature extraction and detection method. Ruan Jian Xue Bao/Journal of Software, 2012,23(2):378−393 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3953.htm [doi: 10.3724/SP.J.1001.2012.03953]

[7] 360 Internet Security Center. Security report of PC in China in 2014. 2015 (in Chinese). http://zt.360.cn/1101061855.php?dtid=1101062360&did=1101205488

[8] Mao WX, Cai ZM, Guan XH, Towsley D. Centrality metrics of importance in access behaviors and malware detections. In: Proc. of the 30th Annual Computer Security Applications Conf. New York: ACM, 2014. 376−385. [doi: 10.1145/2664243.2664286]

[9] Beygelzimer A, Dasgupta S, Langford J. Importance weighted active learning. In: Proc. of the 26th Annual Int'l Conf. on Machine Learning. New York: ACM, 2009. 49−56. [doi: 10.1145/1553374.1553381]

[10] Macskassy SA. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2009. 597−606. [doi: 10.1145/1557019.1557087]

[11] Ji M, Han JW. A variance minimization criterion to active learning on graphs. In: Proc. of the 15th Int'l Conf. on Artificial Intelligence and Statistics. 2012. 556−564.

[12] Zhu XJ, Lafferty J, Ghahramani Z. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: Proc. of the ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining. 2003. 58−65.

[13] Evans I, Long F, Otgonbaatar U, Shrobe H, Rinard M, Okhravi H, Sidiroglou-Douskos S. Control jujutsu: On the weaknesses of fine-grained control flow integrity. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. New York: ACM, 2015. 901–913. [doi: 10.1145/2810103.2813646]

[14] Moser A, Kruegel C, Kirda E. Limits of static analysis for malware detection. In: Proc. of the 23rd Annual Computer Security Applications Conf. IEEE, 2007. 421–430. [doi: 10.1109/ACSAC.2007.21]

[15] Lanzi A, Balzarotti D, Kruegel C, Christodorescu M, Kirda E. AccessMiner: Using system-centric models for malware protection. In: Proc. of the 17th ACM Conf. on Computer and Communications Security. New York: ACM, 2010. 399–412. [doi: 10.1145/1866307.1866353]

[16] Yin H, Song D, Manuel E, Kruegel C, Kirda E. Panorama: Capturing system-wide information flow for malware detection and analysis. In: Proc. of the 14th ACM Conf. on Computer and Communications Security. New York: ACM, 2007. 116–127. [doi: 10.1145/1315245.1315261]

[17] Huang Q, Zeng QK. Taint propagation analysis and dynamic verification with information flow policy. Ruan Jian Xue Bao/Journal of Software, 2011,22(9):2036−2048 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3874.htm [doi: 10.3724/SP.J.1001.2011.03874]

[18] Canali D, Lanzi A, Balzarotti D, Kruegel C, Christodorescu M, Kirda E. A quantitative study of accuracy in system call-based malware detection. In: Proc. of the 2012 Int'l Symp. on Software Testing and Analysis. New York: ACM, 2012. 122–132. [doi: 10.1145/2338965.2336768]

[19] Sun WQ, Sekar R, Liang ZK, Venkatakrishnan VN. Expanding malware defense by securing software installations. In: Proc. of the Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin, Heidelberg: Springer-Verlag, 2008. 164–185. [doi: 10.1007/978-3-540-70542-0_9]

[20] King ST, Chen PM. Backtracking intrusions. ACM Trans. on Compututer System, 2005,23(1):51−76. [doi: 10.1145/1047915.1047918]

[21] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology, 2007,58(7):1019−1031. [doi: 10.1002/asi.20591]

[22] Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000,290(5500):2323−2326. [doi: 10.1126/science.290.5500.2323]

附中文参考文献:

[6] 王蕊,冯登国,杨轶,苏璞睿.基于语义的恶意代码行为特征提取及检测方法.软件学报,2012,23(2):378−393. http://www.jos.org.cn/1000-9825/3953.htm [doi: 10.3724/SP.J.1001.2012.03953]

[7] 360互联网安全中心.2014年中国个人电脑上网安全报告.2015. http://zt.360.cn/1101061855.php?dtid=1101062360&did=1101205488

[17] 黄强,曾庆凯.基于信息流策略的污点传播分析及动态验证.软件学报,2011,22(9):2036−2048. http://www.jos.org.cn/1000-9825/3874.htm [doi: 10.3724/SP.J.1001.2011.03874]

毛蔚轩(1987－),男,辽宁大连人,博士生,主要研究领域为恶意代码分析,数据驱动安全分析.

童力(1991－),男,硕士,主要研究领域为数据挖掘,机器学习.

蔡忠闽(1975－),男,博士,教授,教师,博士生导师,主要研究领域为网络空间安全,人机交互行为分析,数据挖掘.