doi:10.14132/j.cnki.1673-5439.2019.03.012

基于 NE-VASVM 的 JavaScript 恶意代码检测系统

管 衡[1],李麟俊[2],张 琳[1]

(1. 南京邮电大学 计算机学院,江苏 南京 210023
2. 江苏省公安厅 交通警察总队,江苏 南京 210049)

摘要:针对传统的 JavaScript 恶意代码静态检测所存在的样本标记工作量大,以及由于样本冗余度高、泛化能力不足所导致的分类精确度低的问题,提出了一种新的支持向量机的自主学习策略 VASVM,通过价值度量的定义优化了最有价值样本的选择策略,同时结合迭代地调整训练集平衡度,提高了训练集泛化能力和训练过程的收敛速度。然后在此基础上利用 NE-SVM 算法对采用 VASVM 所选择的训练集进行剪裁以降低样本冗余度,并且进一步提高了泛化能力。最后得到了基于 VASVM 策略和 NE-SVM 算法所结合形成的 NE-VASVM 系统。实验结果表明,基于 NE-VASVM 的 JavaScript 恶意代码检测系统有效减少了人工标记工作量,提高了分类器精度。

关键词:支持向量机;主动学习;价值度量;训练集剪裁

中图分类号:TP393 **文献标志码**:A **文章编号**:1673-5439(2019)03-0082-09

JavaScript malicious code detection system based on NE-VASVM

GUAN Heng[1], LI Linjun[2], ZHANG Lin[1]

(1. School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
2. Department Traffic Police Corps, Jiangsu Provincial Public Security, Nanjing 210049, China)

Abstract: Aiming at the problem that the traditional JavaScript malicious code static detection has a large workload of sample tags, and because of high sample redundancy, insufficient generalization ability and low classification accuracy, a new support vector machine is proposed. The self-learning strategy VASVM optimizes the selection strategy of the most valuable samples through the definition of value metrics. At the same time, it adjusts the balance of the training set iteratively, which improves the generalization ability of the training set and the convergence speed of the training process. Then based on this, the NE-SVM algorithm is used to tailor the training set selected by VASVM to reduce the sample redundancy and further improve the generalization ability. Finally, the NE-VASVM system based on VASVM strategy and NE-SVM algorithm is obtained. The experimental results show that the JavaScript malicious code detection system based on NE-VASVM effectively reduces the workload of manual marking and improves the accuracy of the classifier.

Keywords: support vector machines (SVM); active learning; value measure; training set tailoring

收稿日期:2019-03-21 **本刊网址**:http://nyzr.njupt.edu.cn

基金项目:国家自然科学基金(61402241)、江苏省高校自然科学研究项目(17KJB520026)和南京邮电大学校级科研基金(NY217050)资助项目

作者简介:管衡,男,硕士研究生;张琳(通讯作者),女,博士,副教授,zhangl@njupt.edu.cn

引用本文:管衡,李麟俊,张琳. 基于 NE-VASVM 的 JavaScript 恶意代码检测系统[J]. 南京邮电大学学报(自然科学版),2019,39(3):82-90.

在互联网高速发展的今天,网络技术的提升与丰富使得越来越多的网站开始以 Web 应用的形式提供服务,从而导致基于 Web 的应用呈现倍数级的增长。而 JavaScript 作为一种具有完备功能的语言,被广泛地应用于 Web 应用的前端开发之中。Bichhawat 等[1]的研究结果显示95%以上的 Web 站点在进行 Web 前端开发时选用 JavaScript 语言。JavaScript 语言具有跨平台性,可远程嵌入,能够被动态执行[2]。虽然为用户带来了诸多便利和良好的交互体验,但与此同时也给 Web 用户终端带来了不少威胁与风险[3-4]。为了应对 JavaScript 恶意代码所带来的网络安全问题,学术界已经提出过一些检测方法,针对恶意 JavaScript 代码的传统静态检测通常是对样本进行分类标记后,提取特征向量训练生成分类器来对新的未知样本进行预测分类。然而实际成果却不尽人意,存在诸多不足:

(1) 在脚本的特征提取方面,传统的检测系统没有充分地考虑到混淆代码特征,导致特征纬度低,泛化性小。

(2) 近年来,利用未标记实例进行学习越来越受到关注,其主流技术之一就是主动学习[5-8]。主动学习的核心要素在于如何利用有限的标记代价得到高质量的标记数据来提高性能[9]。然而以往的检测系统所使用的主动学习算法大多仅根据未标记样本与超平面之间的距离来对样本价值进行评判,通常认为距离分类超平面越近的样本点越不确定,也就越具有信息价值,最有可能改变分类超平面的位置,而距离越远的样本点则越没用,对位置的改变没有足够的影响力,所以传统的主动学习算法采样策略是迭代地选择与超平面距离最近的样本。这样的做法会造成两个问题:① 每次只选择与超平面距离最近的样本,会导致样本的规模过小而难以及时获取未标记样本集的总体特征,会对收敛速度和泛化能力产生负面影响。② 由于是迭代地选择,所以第 n 次与第 $n-1$ 次所选择的最具价值样本可能会产生信息冗余,也会影响到分类器的泛化效果和检测精度。

(3) 在分类算法的选择中,支持向量机由于具有较强的泛化能力而受到人们青睐[10],但是当具体的样本集中两类样本混合重叠比较严重时候,会导致 SVM 的分类面过于繁杂而对泛化能力产生负面影响。

针对上述种种问题,本文提出了一种新的支持向量机的自主学习策略 VASVM,通过价值度量的定义优化了最有价值样本的选择策略,同时结合迭代地调整训练集平衡度,提高了训练集泛化能力和训练过程的收敛速度。然后在此基础上利用 NE-SVM 算法对采用 VASVM 所选择的训练集进行剪裁以降低样本冗余度并且进一步提高了泛化能力。最后得到了基于 VASVM 策略和 NE-SVM 算法所结合形成的 NE-VASVM 系统。在特征提取方面,充分考虑了四种 JavaScript 混淆代码特征,提高了特征维度。

1　相关工作

Angluin[11]将最有利于机器学习提高效能的样本选择出来,再进行人工标记后用以分类模型训练,该方法受到网络安全领域的广泛关注。Almgren 等[12]在入侵系统中引入了主动学习,获取到了较好的检测性能,李洋等[13]将主动学习与 KNN 算法相结合用于入侵检测,减少了样本标记量和算法开销,Gu 等[14]研究了主动学习应用于入侵检测的现状,指出了一些不足和需要改进的方向。

文献[15]中采用 ESVM(Editing Supporting Vector Machines)方法提高泛化能力,但是需要重复使用 SVM 训练进行迭代的样本剪裁,做法十分的复杂。

在脚本的特征提取方面,Likarish 等[16]提取代码中的关键字作为特征,应用机器学习算法进行分类来进行脚本检测,该方法只考虑了脚本特征中关键字的不同分布,并没有考虑一些可执行的方法与函数特征,导致特征类型过于单一,特征维度太低。Al-Taharwa 等[17]通过分析 JavaScript 脚本语义特征,建立抽象语法树来进行分类检测,但并没有考虑到 JavaScript 脚本中混淆代码的特性,因此难以准确检测。Fraiwan 等[18]在脚本的频繁特征、URL 特征、函数特征和执行特征的基础上构建相应的分类模型,亦是缺乏对混淆特征的考虑。马洪亮等[19]使用基于机器学习的分类算法来检测 JavaScript 恶意代码,但是局限于特征提取的维度缺失,所以检测效果一般。

2　NE-SVM 算法

2.1　特征提取

结合以往的文献[20-21],充分考虑了 JavaScript 脚本的混淆特征,基于统计特征、漏洞利用特征和动态执行特征这四类特征,完善了传统的静态检测所存在的特征提取维度低、泛化性不足的问题。

最终确定提取 25 个特征,完善了传统的静态检测所存在的特征提取维度低、泛化性不足的问题,特征分布如下:

(1) eval 函数个数
(2) classid 个数
(3) 字符串包含 iframe 个数
(4) DOM 更改函数个数
(5) Unescape( ) 函数个数
(6) 字符串总信息熵
(7) 字符串最大信息熵
(8) 字符串含参个数
(9) 字符串内嵌脚本长度
(10) 长度大于 50 的字符串个数
(11) 可疑字符串使用个数
(12) 代码混淆个数
(13) Window. setTimeout( ) 函数个数
(14) Shellcode 使用个数
(15) CreatObje ct( ) 函数使用个数
(16) 含有某事件的特征符使用个数
(17) Document. write 或. Write ln( ) 个数
(18) 可疑标签个数
(19) 字符串更改函数个数
(20) 脚本中空格回车所占比例
(21) 十六进制字符个数
(22) parseInt( ) 和 fromcharco de( ) 个数
(23) ActiveXOBbject 相关函数使用个数
(24) 字符串最大长度
(25) Escape( ) 函数个数

在收集到所需的恶意样本与正常样本作为训练集后,再对每个样本 $\boldsymbol{x}_i$ 进行相应的特征提取,将恶意样本特征标记为 +,正常样本特征标记为 i,在对每个 JavsScript 样本的特征数量进行归一化处理之后得到对应的特征向量 $\boldsymbol{x}_i = (x_i^1, x_i^2, \cdots, x_i^{25})$, $x_i^n \in [0,1]$用于 SVM 的分类训练。

2.2 传统 SVM 算法

支持向量机(Support Vector Machine, SVM)算法由 Vapnik 等[22]在 1995 年提出,主要用于解决模式分类问题。支持向量机利用对最优分类超平面的寻找来应对分类问题,即寻找决策面。选择 SVM 只要基于该算法具备如下三个优点[23]:(1) 经验风险最小化且避免分类模型过拟合,(2) 可得到全局最优解,(3) 应用范围广。

SVM 属于二分类模型,根据结构风险最小化准则,构建目标函数使得类别尽可能分开。该模型的基本原理:假设一个包含两类样本的集合$\{(\boldsymbol{x}_i, y_i) \mid i = 1,2,\cdots,l$ 且 $\boldsymbol{x}_i \in R^n, y_i \in \{-1,1\}\}$能够被某个超平面 $\omega * x + b = 0$ 正确分开,其中,R^n 为 n 维实数空间,l 为样本中点的数目。那么将与两类样本点距离最大的分类面称为最优超平面,距离次平面最近的两类样本点被称为支持向量(Support Vector)。支持向量机分为线性可分支持向量机、线性不可分支持向量机以及非线性可分支持向量机三种。

本文的正常与恶意的脚本分类属于非线性可分问题。所以采用非线性可分支持向量机。

SVM 算法需要通过内积核函数来将低维空间中那个非线性问题转为高维空间线性划分问题求解。向量机支持多种核函数,对于核函数的选取,采用了线性核函数:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i * \boldsymbol{x}_j \tag{1}$$

选取线性核函数主要是基于以下两点考虑:

(1) 本文所选取的特征较之以往要多出很大部分,选取参数会耗时太长,而线性核函数只有一个惩罚因子,选取线性核函数有助于优化参数的选取时长。

(2) SVM 优势在于对支持向量的取用以减少 H 空间,减少过拟合的可能性,所以在 H 特征向量维数比较大的情况下,线性核函数更加适用。

2.3 改进后算法 NE-SVM

在对分类器进行训练时,传统的 SVM 着眼点在于两类样本交界部分,有些点混杂在另外一类中,不仅无助于分类器性能的提高,反而会增加训练器的计算负担,也由于这些点的存在,会造成训练器过度学习,减弱了泛化能力。

针对以上问题,对传统的 SVM 进行改进,提出了另外一种泛化能力提高的算法 NE-SVM(Nearest neighbor-Editing-SVM),对每一个样本,将其与最近邻的样本进行异同的比较然后决定其取舍,以此来进行对训练集的剪裁。NE-SVM 迭代地寻找每个点的最近邻近点,保留与该点相同的同类,删除不同的异类,从而避免了同类样本过度重叠。提高了泛化性。其中的距离计算利用欧式距离作为两个向量之间的距离计算,设:

$$\boldsymbol{x}_i = (x_i^1, x_i^2, \cdots, x_i^n), \boldsymbol{x}_j = (x_j^1, x_j^2, \cdots, x_j^n) \tag{2}$$

则样本 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 间的欧式距离为:

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{k=1}^{n} (x_i^k - x_j^k)^2} \tag{3}$$

算法描述如下:

算法 1　NE-SVM 算法

输入:训练集$(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)$

1. $\boldsymbol{Train}_{m*n}=[\boldsymbol{X},\boldsymbol{Y}]^{\mathrm{T}},\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m]^{\mathrm{T}}$
2. $\boldsymbol{Y}=[y_1,\cdots,y_m]^{\mathrm{T}}$
3. for $p=1$ to m
4. $\{\boldsymbol{Z}_{1*m}=(z_{ij}),z_{ij}=\infty,i=1$
5. $j=1,2,\cdots,m$
6. for $q=1$ to m
7. $\{$if $q\neq p,z_{lq}=D(x_p,x_q);\}$
8. $\}$找出任一点与其他点距离,与自身距离无穷。
9. $\boldsymbol{NN}_{m*1}=(nn_{ij}),nn_{ij}=1,i=1,2,\cdots,m,j=1$
10. $t=1$;value $=z_{11}$
11. for $q=1$ to m
12. $\{$if $z_{1q}<$value$\{$value $=z_{1q}$;$t=q$;$\}$
13. $n_{p1}=t;\}$找出最短距离及对应点
14. $\boldsymbol{L}_{m*1}=(l_{ij}),l_{ij}=1,i=1,2,\cdots,m,j=1$
15. for $p=1$ to m
16. $\{$if $y_p\neq y_{nn_p},l_{p1}=-1;\}$

判断每个向量的类标与最近领是否一致并分别记为 1 与 -1。

17. $i=0$
18. for $p=1$ to m
19. $\{$if $l_{(p-i)1}=-1$
20. $\{$删除矩阵 $\boldsymbol{TR}$ 及 $\boldsymbol{L}$ 的 $p-i$ 行,新矩阵仍为 $\boldsymbol{TR}$ 及 $\boldsymbol{L}$;$i=i+1;\}$
21. $\}$

输出:修剪后的训练集 $\boldsymbol{TR}$

3　NE-VASVM 系统

传统的分类检测对于训练样本的需求量十分巨大,而对于样本的标注往往是实验者根据先验知识进行人工标记,时间成本和人力消耗都很大。其次,当今时代恶意代码攻击层出不穷,日渐繁杂,对于很多新型的恶意代码的出现,则需要时常更新训练模型,不断地分析新的攻击,进行再次标记,训练新的模型,对于阻止恶意代码的扩散不具备及时性。针对以上问题,通过将改进后的主动学习算法加入到 NE-SVM 的样本选择之中形成 NE-VASVM 系统,利用询问机制,选择对于提高分类器效率最有利的样本,用相对更少的训练样本得到更高效的分类效率,减少标注样本所需代价,达到事半功倍的效果。

3.1　传统的主动学习策略

对于选择引擎而言,用于选择标注最具价值信息的询问机制是最核心的要素,而采用何种原则选取最有价值样本进行标注则是主动学习的关键所在。

传统的主动学习算法中:设存在样本集 $\phi=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\},\boldsymbol{x}_i\in R^n,i=1,2,\cdots,n,\boldsymbol{x}_i$ 表示样本中第 i 个样本的特征向量,$y_i\in\{+1,-1\}$ 为分类识别号,假设是线性可分的样本集合,则存在一个能够分离两个类别的超平面的决策方程:

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0,\boldsymbol{x}\in R^n \tag{4}$$

任一样本$(\boldsymbol{x}_i,y_i),i=1,2,\cdots,n$,都满足:

$$\begin{cases}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\geqslant+1,y_i=+1\\ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\leqslant-1,y_i=-1\end{cases} \tag{5}$$

则在 R^n 中任一样本 $\boldsymbol{x}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)^{\mathrm{T}}$ 到分类平面的距离:

$$d=\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_j+b}{\|\boldsymbol{w}\|},\text{其中}\ \|\boldsymbol{w}\|=\sqrt{\boldsymbol{w}^{\mathrm{T}}\cdot\boldsymbol{w}} \tag{6}$$

据式(6)可得:

$$d(\boldsymbol{x})=\boldsymbol{x}\cdot\boldsymbol{w}^*+b^*=\sum_{i=1}^{l}y_i\alpha_i^*(\boldsymbol{x}\cdot\boldsymbol{x}_i)+b^* \tag{7}$$

其中,b^* 是分类域值,可以通过任一对支持向量取中而得到。

对于待测样本 $\boldsymbol{x}'$的判断依据如下:

(1) 当 $d(\boldsymbol{x}')\geqslant1$ 时,则表示 $\boldsymbol{x}'$属于正样本;

(2) 当 $d(\boldsymbol{x}')\leqslant-1$ 时,则表示 $\boldsymbol{x}'$属于负样本;

(3) 当 $-1<d(\boldsymbol{x}')<1$ 时,则表示 $\boldsymbol{x}'$存在于超平面分类间隔之中,属性是不确定的,不确定性正比于 $d(\boldsymbol{x}')$对于 0 的趋近程度。

综上所述,在传统的主动学习算法中,通常认为,离分类超平面越近的样本,因为不确定性大,所以最有可能被分错。信息量最大,最具有选择信息。但如果仅仅以此为标准,那么将会存在两个不足之处:

(1) 每次只考虑距离超平面最近的样本,导致训练集规模太小,无法及时获取无标记样本集的整体特征,影响收敛速度。

(2) 第 n 次迭代中选择的样本可能会于第$n-1$次迭代中选择的样本产生信息冗余。会在一定程度上对训练过程的分类速度和分类效果产生负面影响。

3.2　改进后的主动学习策略 VASVM

针对以上不足之处,对传统的主动学习算法进行改进,定义了价值度量,形成了以价值度量为准则的主动学习策略 VASVM (Value Measure Active SVM)。不仅要对未标记样本与超平面样本的距离进行考量,同时也加入了对未标记样本和已标记样本距离的考虑,从而选出兼备价值以及与之前标记

样本较低冗余度的未标记样本,并且在迭代过程中,调整训练样本集和平衡度,从而获取到相对较快的收敛速度与泛化能力。

3.2.1 定义价值度量

将分类训练过程视为迭代过程,每次迭代都通过价值度量来选取最具价值的未标记样本进行人工标注,然后加入分类器训练器,循环整个过程直到循环次数达到上限或者分类器准确率达到阈值。

假设 M 与 N 分别表示已标记样本集与未标记样本集,对每个未标记样本 $\boldsymbol{x}_i$ 定义如下价值度量:

$$c(\boldsymbol{x}_i)=\frac{\overline{d^n(\boldsymbol{x}_i,\boldsymbol{x}_j)}}{D^n(\boldsymbol{x}_i)},\boldsymbol{x}_i\in M,\boldsymbol{x}_j\in N \tag{8}$$

其中,$\overline{d^n}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 为已标记样本 $\boldsymbol{x}_j$ 与未标记样本 $\boldsymbol{x}_i$ 在第 n 次迭代时候的欧式距离均值。$D^n(\boldsymbol{x}_i)$ 表示未标记样本 $\boldsymbol{x}_i$ 与当前分类超平面在第 n 次迭代中的距离,由式(8)可知,样本的价值度与 $D^n(\boldsymbol{x}_i)$ 成反比,同样$\overline{d^n}(\boldsymbol{x}_i,\boldsymbol{x}_j)$越大,则代表样本信息的冗余度越小,样本的选取价值也就越大。综上所述,价值度量$c(\boldsymbol{x}_i)$越大,则样本的价值越高。在分类训练的迭代过程中,首先对所有的未标记样本计算其价值度量并进行降序排列,然后提取前 k 个样本在进行标注之后加入到训练集之中。

3.2.2 样本集平衡度调整

在迭代后,可能会出现超平面与两类样本间距不同的不平衡情况,所以在按照价值度量对样本进行选择时,可能会存在某一类样本数量远大于另外一类样本数量的情况,如果不对这样的不平衡状态进行数据集处理,会给算法的泛化能力带来负面影响,所以通过对每次迭代后的平衡度 b 的检测来避免最有价值样本选择带来的不平衡状态。b 的定义:$\text{num}^+\leqslant\text{num}^-$ 时 $b=\frac{\text{num}^+}{\text{num}+1}$,否则 $b=\frac{\text{num}^-}{\text{num}^+}$。其中,$\text{num}^+$代表了正类样本数量,$\text{num}^-$则是负类样本数量,预设一个平衡参数 ε,当 $b\leqslant\varepsilon$ 时,则集合视作非平衡的,则需对占据多数的类别数据进行聚类,再将聚类中心与少数量的类别样本加入到训练集之中,删除多余的占据多数的类别样本,以消除训练集的不平衡状态。

3.3 NE-VASVM 系统结构

将 NE-SVM 算法和 VASVM 策略进行结合运用,采用改进后的主动学习策略 VASVM 进行样本的选择,再经过 NE-SVM 所提供的算法进行样本剪裁后将最终的训练样本投入到分类器训练中,形成了 NE-VASVM 系统,系统结构流程如下:

算法 2 NE-VASVM 学习算法

输入:M 为空集,代表已标记样本集,N 为未标记样本集,Need_label 为最有价值样本集合,初始值为 ϕ,且每次循环进入迭代前要进行清空,Wrong_label 为错误的样本,初值为 ϕ,迭代前需清空,Train 为人工标记样本,初值为 ϕ,用于训练 SVM 分类器。

Step1:对 N 中左右样本进行聚类,形成 K 类样本,相应聚类中心为 $c_1,c_2,\cdots,c_k$;然后将 $c_1,c_1,\cdots,c_k$ 进行人工标记,若包含正负类,则令 Train = $\{c_1,\cdots,c_k\}\cup$Train;否则,继续聚至 $k+1$ 类,重复直至包含正负样本后,令 $U=U-$Train。

Step2:循环执行 i 次以下操作。

Step3:将 Train 进行样本剪裁后训练 SVM(采用 NE-SVM 算法),并对 U 中样本集合进行分类预测。

Step4:对于任一 $\boldsymbol{x}_i(\boldsymbol{x}_i\in U)$,计算其价值度量 $c(\boldsymbol{x}_i)$,再降序排列后取前 m 个样本加入 Need_label,并且进行人工标记。

Step5:将 Need_label 与 Step3 进行标签对比,不同者放入 Wrong_label。

Step6:按公式计算 Wrong_label 对应的 b 值,若 $b\leqslant\varepsilon$ 则对 Wrong_label 平衡度进行相应调整,否则转至 Step7。

Step7:令 Train = Wrong_label$\cup$Train,$U=U-$Wrong_label。

Step8:若算法精度达到预设阈值或样本耗尽,则算法结束,否则转至 Step3。

输出:分类结果。

4 实验验证

实验中使用的 JavaScript 脚本是从真实网络环境中收集的。实验中 m 取值为 40。

4.1 数据准备

从 Alexa 所公布的 TOP500 正常网站抓取得到 2 500个正常样本。从知名恶意站点发布站点(PhishTank)与网络病毒数据库 VXHeavens 公布的恶意网站爬虫收集到2 500个 JavaScript 恶意脚本作为恶意样本集。将 500 个正常样本与 500 个恶意样本作为训练样本集,将2 000个正常样本与2 000个恶意样本作为预测样本集,数据分布和测试环境分别如表 1 和表 2 所示。

表 1 数据分布

训练样本	测试样本
1 000	4 000

表 2 测试环境

名称	配置
CPU	5-7300HQ
内存	8 GB
操作系统	Windows 10
实验工具	Python 3.6 Weka

4.2　测试指标

在检测系统的实验中,需要对检测结果进行分析,使用了正负样本准确率 A(Accuracy)、负样本精确率 P(Precision)、召回率 R(Recall),以及综合评价指标 $F1$(F1-Measure)四个指标作为衡量检测效果的衡量标准。

$$A = \frac{TP + FN}{TP + TN + FP + FN} \times 100\% \tag{9}$$

$$P = \frac{FN}{FN + TN} \times 100\% \tag{10}$$

$$R = \frac{FN}{FP + FN} \times 100\% \tag{11}$$

$$F1 = \frac{2PR}{P + R} \tag{12}$$

其中,TP,TN,FN,FP 分别代表每类样本数目,意义如表 3 所示。

表 3　参数含义

参数	含义	说明
TP	Ture Positive	正样本被预测为正样本
TN	Ture Negative	正样本被预测为负样本
FP	False Positive	负样本被预测为正样本
FN	False Negative	负样本被预测为负样

4.3　结果分析

首先为了研究主动学习算法是否真的能够在标记样本数量少的情况下,能够显著提升分类器的各项性能。在样本标记数目等同的情况下,对同样进行样本剪裁后的基于随机采样的分类系统 NE-SVM 和基于改进后的主动学习算法的 NE-VASVM 分类系统,对比两者在采取不同的初始训练集尺寸时的分类效果。分别对初始训练集尺寸为总的训练样本的 10%,20%,30%,40%,50%,60%,70% 七种情况进行对比分析,由于查准率和召回率两个指标相互制约,所以重点采用了综合指标 $F1$ 和准确率 A 作为测评指标。对比结果如图 1 和图 2 所示。

NE-VActiveSVM 系统和 NE-SVM 系统的对比实验证明了主动学习方法的有效性:(1) 相比于随机采样的分类方法,采用基于价值度量采样的主动学习方法能够显著提高分类的各项性能指标,尤其是在标记样本总数相对较少情况下,提升更为明显。(2) 采用基于价值度量采样的主动学习方法在训练集大小只有 50% 左右就可以达到接近峰值的检测效果。与训练集尺寸增加到 60% 和 70% 时并无明显差别。而传统的随机采样分类方法检测效果会随着训练集尺寸的增大有较为明显的提升。要达到接近于主动学习检测效果,峰值所需训练集尺寸超过了 70%。上述结果说明主动学习可以减少传统的分类检测所需要的样本标记工作量,能够实现合理、高效地选择训练样本,能够在得到较高检测效果的前提下大幅降低所需训练样本数量。

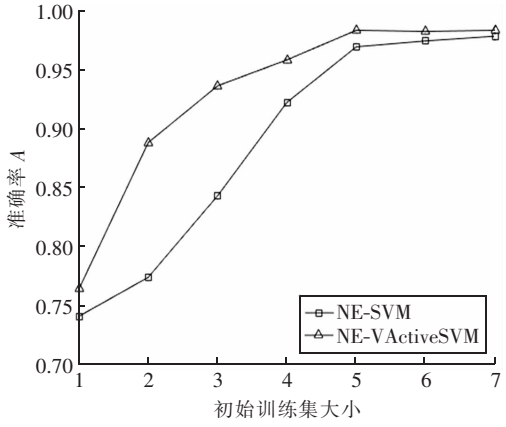

图 1　准确率对比 1

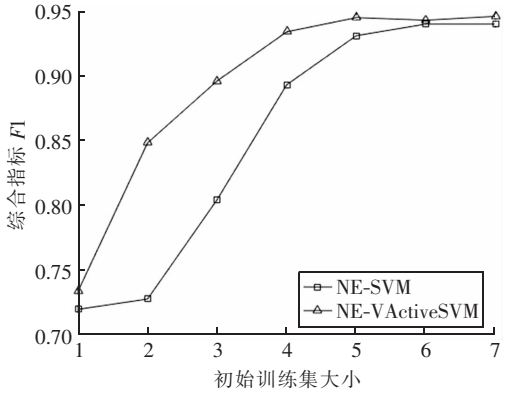

图 2　综合指标对比 1

其次,为了测试本文提出的模型,通过对传统主动学习算法的改进和样本剪裁策略的加入,最终实现更有效的 JavaScript 恶意代码检测效果。采用控制变量法进行了四组系统的对比实验,所有系统均采用 50% 的训练集尺寸训练生成分类器,然后将预测样本随机分成 10 份投入到训练生成的分类器,然后记录 10 次的各项指标,进行对比的四个分类系统分别是:

(1) ZeroSVM 系统:基于随机采样的无样本剪裁的传统 SVM 静态检测系统。

(2) ActiveSVM 系统:基于传统自主学习算法的 SVM 静态检测系统。

(3) VActiveSVM 系统:基于改进后 VActive-SVM 的静态检测系统。

(4) NE-VActiveSVM 系统:加入 NE-SVM 样本剪裁后的 VActive-SVM 静态检测系统。

实验得到基于测试指标准确率 A 的结果对比如图 3 所示。

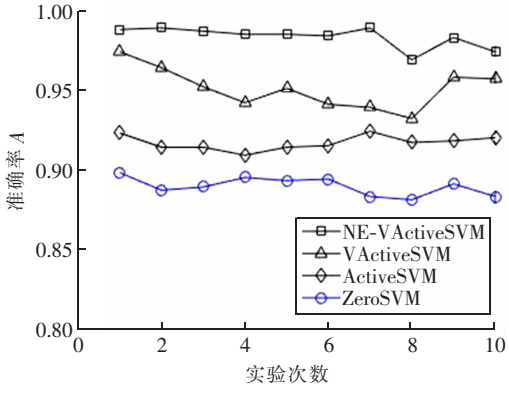

图 3 准确率对比 2

基于测试指标查准率 P 的结果对比如图 4 所示。

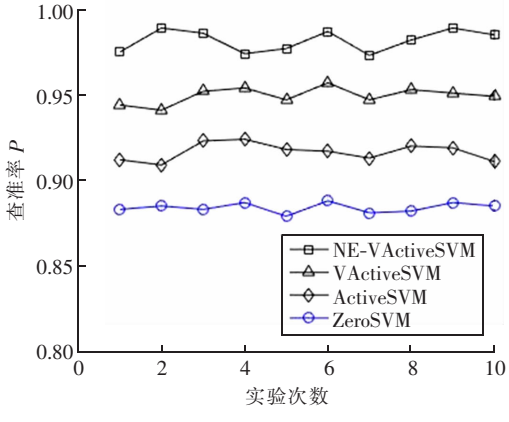

图 4 查准率对比

基于测试指标召回率 R 的结果对比如图 5 所示。

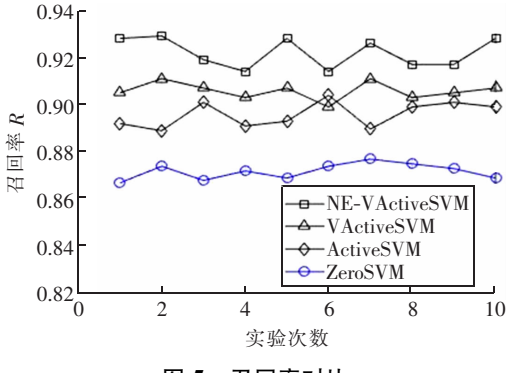

图 5 召回率对比

基于测试指标综合指标 $F1$ 的结果对比如图 6 所示。

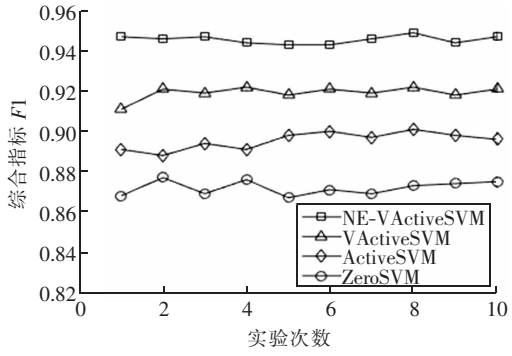

图 6 综合指标对比 2

由图 3 ~ 图 6 中 ActiveSVM 和 VActiveSVM 两个系统的实验结果对比分析可以得到,改进后的主动学习算法相对于传统的主动学习算法在四个性能指标上都有更为优秀的表现,实现了对于传统的自主学习算法的优化和改进。

由图 3 ~ 图 6 中 NE-VActiveSVM 系统和 VActiveSVM 两个系统的实验结果对比分析可以得到,样本剪裁策略能够有效地提升分类系统的检测效率。

最后与 Likarish[16], Jodavi 等[24] 以及 Wang[25] 的 JavaScript 代码检测方法进行了对比,使用综合评价指标 $F1$、查准率 P 与召回率 R 来衡量检测结果,对比结果如图 7 所示。

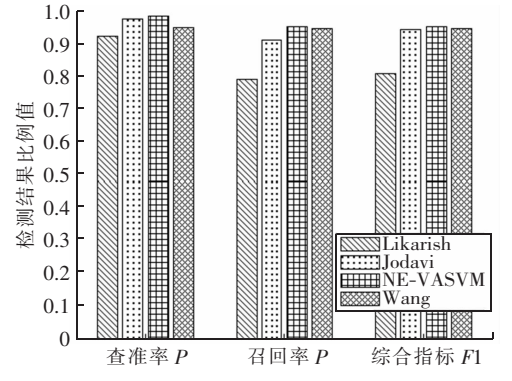

图 7 文献对比

由 NE-VASVM 系统和 ZeroSVM 系统以及相对于文献[16,24 -25]等的静态检测方法的检测结果对比得到:相比于传统的静态检测系统,NE-VASVM 系统能够显著提升传统静态检测系统的各项性能指标,实现了检测系统的优化。相比于文献[16,24 -25]等的静态检测方法,本文所提出的方法也更具优势。

5 结束语

论文分别对主动学习策略和支持向量机算法进行了改进,并在此基础上探讨了 JavaScript 恶意代码

检测系统。但仍然存在静态检测系统对于动态执行的恶意代码检测效果较差的问题;在特征提取方面仍然存在特征维度不高,权重分配不够科学的问题,在后续的工作中均会加以完善。

参考文献:

[1] BICHHAWAT A, GARG D. Information flow control in WebKit's JavaScript bytecode[C]//3rd Conference on Principles of Security and Trust(POST). 2014:159 – 178.

[2] ZHOU Y, EVANS D. Understanding and monitoring embedded Web scripts[C]//Proceedings of the IEEE Symposium on Security and Privacy. 2015:850 – 865.

[3] BROWN F, NARAYAN S, WAHBY R S, et al. Finding and preventing bugs in JavaScript bindings[C]//IEEE Symposium on Security and Privacy (SP). 2017.

[4] HEDIN D, SABEFELD A. Web application security using JSFlow [C]//Proceedings of the 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing(SYNASC). 2015:16 – 19.

[5] ZHANG X Y, WANG S, YUN X. Bidirectional active learning: a two-way exploration into unlabeled and labeled dataset[J]. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(12):3034 – 3044

[6] 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[J]. 软件学报,2017,28(2):384 – 397.
MAO Weixuan, CAI Zhongmin, TONG Li. A method of malicious code detection based on active learning[J]. Journal of Software, 2017, 28(2):384 – 397. (in Chinese)

[7] 高飞,高新波. 主动特征学习及其在盲图像质量评价中的应用[J]. 计算机学报,2014,37(10):2227 – 2234.
GAO Fei, GAO Xinbo. Active feature learning and its application in blind image quality evaluation[J]. Chinese Journal of Computers, 2014, 37(10): 2227 – 2234. (in Chinese)

[8] 刘敬,谷利泽,钮心忻,等. 基于单分类支持向量机和主动学习的网络异常检测研究[J]. 通信学报,2015,36(11):136 – 146.
LIU Jing, GU Lize, NIU Xinxin, et al. Research on network anomaly detection based on single-class support vector machine and active learning[J]. Journal on Communications, 2015, 36(11):136 – 146. (in Chinese)

[9] DU B, WANG Z, ZHANG L, et al. Exploring representativeness and informativeness for active Learning[J]. IEEE Transactions on Cybernetics, 2015, 47(1):14 – 26.

[10] 武小年,彭小金,杨宇洋,等. 入侵检测中基于 SVM 的两级特征选择方法[J]. 通信学报,2015,36(4):23 – 30.
WU Xiaonian, PENG Xiaojin, YANG Yuyang, et al. A two-level feature selection method based on SVM in intrusion detection[J]. Journal on Communications, 2015, 36(4): 23 – 30. (in Chinese)

[11] ANGLUIN D. Queries and concept learning [J]. Machine Learning, 1988, 2(4):319 – 342.

[12] ALMGREN M, JONSSON E. Using active learning in intrusion detection[C]//Proceedings of 17th IEEE Computer Security Foundations Workshop. 2004:88 – 98.

[13] 李洋,方滨兴,郭莉,等. 基于主动学习和 TCM-KNN 方法的有指导入侵检测技术[J]. 计算机学报,2007,30(8):1464 – 1473.
LI Yang, FANG Binxing, GUO Li, et al. Guided intrusion detection technology based on active learning and TCM-KNN method[J]. Chinese Journal of Computers, 2007, 30(8): 1464 – 1473. (in Chinese)

[14] GU Y J, ZYDEK D. Active learning for intrusion detection [C]//National Wireless Research Collaboration Symposium. 2014:117 – 122.

[15] KE Haixin, ZHANG Xuegong. Editing support vector machines[C]//Proceedings of International Joint Conference on Neural Networks. 2001:1464 – 1467.

[16] LIKARISH P, JUNG E, JO I. Obfuscated Malicious JavaScript detection using classification techniques [C]//4th International Conference on Malicious and Unwanted Software(MALWARE). 2009:47 – 54.

[17] AL-TAHARWA I A, LEE H M, JENG A B, et al. Redjsod: a readable JavaScript obfuscation detecto using semantic-based analysis [C]//11th International Conference on Trust, Security and Privacy in Computing and Communications. 2012:1370 – 1375.

[18] FRAIWAN M, AL-SALMAN R, KHASAWENEH N, et al. Analysis and identification of Malicious JavaScript code [J]. Information Security Journal: A Global Perspective, 2012, 21(1):1 – 11.

[19] 马洪亮,王伟,韩臻. 混淆恶意 JavaScript 代码的检测与范混淆研究[J]. 计算机学报,2017,40(7):1699 – 1713.
MA Hongliang, WANG Wei, HAN Zhen. Research on detection and Van confusion of confusing malicious JavaScript code[J]. Chinese Journal of Computers, 2017, 40(7):1699 – 1713. (in Chinese)

[20] SONI P, BUDIATO E, SAXENA P. The SICILIAN defense: signature-based white listing of Web JavaScript [C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015:1542 – 1557.

[21] WU H, QIN S. Detecting obfuscated suspicious JavaScript based on collaborative training[C]// 17th International Conference on Communication Technology (ICCT). 2017.

[22] VAPNIK V N. The Nature of Statistical Learning Theory [M]. Berlin: Springer, 2000.

[23] CRISTIANINI N, SHAWE-TAYLOR J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods [M]. Cambridge: Cambridge University Press, 2000.

[24] JODAVI M, ABADI M, PARHIZKAR E. JSObfusDetector: a binary PSO-based one-class classifier ensemble to detect obfuscated JavaScript code [C]// The International Symposium on Artificial Intelligence and Signal Processing (AISP). 2015: 322 - 327.

[25] WANG Y, CAI W D, WEI P C. A deep learning approach for detecting malicious javascript code [J]. Security & Communication Networks, 2016, 51(8): 28656 - 28667.

声　明

为适应我国信息化建设的需要,扩大作者学术交流渠道,实现期刊编辑、出版工作的网络化,本刊已加入《中国学术期刊(光盘版)》、《中国期刊网》全文数据库、《万方数据——数字化期刊群》和《中文科技期刊数据库》,并已许可《中国学术期刊(光盘版)》电子杂志社在中国知网及其系列数据库产品中,以数字化方式复制、汇编、发行、信息网络传播本刊全文,作者著作权使用费随本刊稿酬一次性给付。如不同意将文章编入相关数据库,请在来稿时声明,本刊将做适当处理。

本刊编辑部