一种改进主动学习的恶意代码检测算法

李翼宏 刘方正 杜镇宇
(国防科技大学电子对抗学院 合肥 230037)

摘 要 传统的恶意代码检测技术依赖于大量的已标记样本,然而新出现的恶意代码的标记数量往往较少,使得传统的机器学习检测方法难以取得较好的检测效果。针对该问题,研究了一种改进主动学习的恶意代码检测算法,提出了基于最大距离(Maximum Distance)的样本选择策略和基于最小估计风险(Minimum Risk Estimate)的样本标记策略,实现了已标记样本较少情况下的恶意代码检测。实验结果显示,相比于未使用主动学习的方法,该算法的总体检测效果更好,在已标记样本数量占比为10%的情况下,其比随机选择策略的主动学习的效果更好,在时间性能上比人工标记策略的主动学习效果更好。

关键词 主动学习,恶意代码,特征,估计风险,标记
中图法分类号 TP393.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.05.014

Malware Detection Algorithm for Improving Active Learning

LI Yi-hong LIU Fang-zheng DU Zhen-yu
(Electronic Countermenaure Institute, National University of Defense Technology, Hefei 230037, China)

Abstract The traditional malware detection technology relies on a large number of labeled samples. However, the number of marked labels is often less for the new malwares, so the traditional machine learning detection methods are difficult to get good detection results. Therefore, this paper proposed a malware detection algorithm based on active learning. It contains a sample selection strategy based on Maximum Distance and a sample tagging strategy based on Minimum Risk Estimate, which can achieve better detection results with a small number of marked samples. Experimental results show that the proposed algorithm performs better than the overall detection method without active lear-ning, and the active learning effect is better when the number of labeled samples is 10% compared with the random selection strategy. Moreover, the algorithm has better temporal performance than the active learning strategy of artificial tagging strategy.

Keywords Active learning, Malware, Features, Estimated risk, Sample

1 引言

2017年以来,恶意代码数量依然呈上升的趋势,尤其是新型恶意代码,其数量始终呈逐年递增状态,这对网络空间安全造成了极大的威胁[1]。随着技术的不断发展,机器学习能够自动发掘恶意代码的内在规律并不断完善自身的性能,进而实现对未知恶意代码的检测。因此,相比于传统的基于特征码、hash值等的检测技术,采用基于机器学习的方法对恶意代码进行建模和检测时具有更高的检测率,其已成为恶意代码检测领域的研究热点。在机器学习的训练阶段,往往需要大量的、完备的恶意代码样本集才能达到理想的检测效果。然而在现实网络环境中,一方面,恶意代码样本,特别是新型恶意代码样本,数量较少,难以形成完备的训练集,从而影响了恶意代码的检测效果;另一方面,对恶意代码进行分析和标注也需要大量的人力和物力,从而影响了恶意代码的检测效率。因此,如何在小规模恶意代码样本的情况下实现较为理想的检测效果和效率,是恶意代码检测领域研究的重点和难点。

现实情况中,在新型恶意代码出现之初,被标记的恶意样本数量较少,难以训练泛化能力较强的检测模型,若采用人工的方式对未知的恶意样本进行标记,则需要大量的人力和物力。为解决该问题,主动学习应运而生,其可以利用当前小样本训练的检测模型,主动地选择其中最有价值的样本进行标记,然后再将该样本加入到之前模型的训练集中重新训练,以不断提高检测模型的泛化能力。主动学习可以显著地减少训练所需的样本数量。

2 相关工作

在现有的研究中,Tong等[2-3]提出了一种基于支持向量机(Support Vector Machine,SVM)的主动学习方法,即将最

到稿日期:2018-04-26 返修日期:2018-08-15 本文受国家自然科学基金(U1636201)资助。
李翼宏(1994-),男,硕士,主要研究方向为网络信息安全,E-mail:norrislee22@gmail.com;**刘方正**(1982-),男,博士,副教授,主要研究方向为计算机应用技术、人工智能和通信技术,E-mail:yoyofangzheng@aliyun.com(通信作者);**杜镇宇**(1996-),女,硕士,主要研究方向为网络攻击防御技术。

靠近 SVM 分类面的样本作为不确定样本，对其进行人工标记并加入原训练集，然后重新训练分类模型。陈耀东等[4]在 SVM 的基础上引入了一种直推式支持向量机的方法（Transductive Support Vector Machines，TSVM[5]），其利用待分类样本的内部信息来提高学习器的局部分类性能。实验结果表明，在少标注样本少于 400 的情况下，TSVM 比 SVM 约高出 4%～6%的性能。另一种主动学习方法是由 Seung 等[6]和 Freund 等[7]提出的基于委员会投票（Query-By-Committee，QBC）选择的主动学习技术，其通过训练多个分类器，将投票不一致的样本进行标记并重新加入训练集，能够使用很少的样本达到理想的检测精度。相比于 SVM，该方法的速度更快。但是上述方式在选择未标注的样本后，都采用人工标注的方式对样本进行标记，时间开销较高，不利于快速形成检测。毛蔚轩等[8]通过计算样本的相似性，再结合最小化估计风险的主动学习来选择样本，省去了人工标注的环节，直接将相似度较高的样本加入原来的训练集，在小规模恶意代码样本的情况下降低了错误率。但是，该方法的局限性较大，需要构建数据依赖网络并找到重要资源对象，不具有通用性。

基于上述分析，本文对主动学习的选择策略和标记策略进行改进，利用样本自身信息和分类器自动标记未知样本，提出一种最大特征距离选择策略和最小估计风险标记策略的主动学习算法。该方法选择性地对样本进行标记且不依赖于人工，进一步提高了主动学习的效率，能够在恶意代码样本数量较少的情况下不断增量训练，逐渐提高模型的泛化能力，从而达到较为理想的检测效果，为新型恶意代码的检测提供了支持。

3　本文算法的基本思想

本文的目的是在少量已标记样本条件下提高对未标记样本的检测率。所提算法的基本思想是：首先将有恶意代码标签和正常代码标签的所有样本作为已标记样本集 L，将没有标签的所有未知样本作为未标记样本集 U；然后提取所有样本特征，并利用 simhash 算法[9]对特征进行规范化处理，将其表示成统一的格式；其次将 L 的特征作为输入，训练分类器 C；接着将 U 的特征作为输入，利用基于最大特征距离的样本选择策略对 U 进行选择，并将选择出的样本放入待标记样本集 S；最后利用基于最小估计风险的样本标记策略对 S 中估计风险值最低的样本进行标记，并将标记后的样本加入 L；更新 L 和 U，并重新对 C 进行训练，直到 U 中所有样本被标记完毕。该算法的基本流程如图 1 所示。

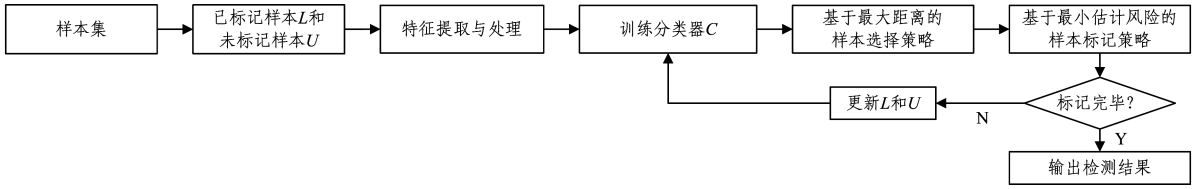


图 1　改进主动学习的恶意代码检测算法的流程

Fig. 1　Flow of malware detection algorithm for improving active learning

3.1　特征的提取与处理

在恶意代码检测中，样本的特征提取和处理是后续建模检测的关键，本文主要将样本的 API 调用函数作为特征提取的对象。API 函数是恶意代码实现其恶意行为并与系统交互所必须的函数，虽然 API 本身是没有恶意性的，但是恶意代码通过某些 API 函数的组合，可使其所表示的行为构成恶意性，而这些行为在正常文件中是不常见的，如进程的注入操作、关键系统文件的更改和删除等[10-11]。因此，本文对 API 函数的调用序列 $X_i=\{api_1 api_2 \cdots api_n\}$ 进行提取，其中，$i(i\in L\cup U)$ 表示第 i 个样本，n 为 API 函数的数量。

由于序列的长度不一，将其直接作为特征会增加后续建模的计算复杂度，从而影响检测的效果，因此本文采用 simhash 算法对该序列进行处理，将每个特征都表示成相同位数的二进制形式。图 2 显示了 simhash 算法的处理过程。

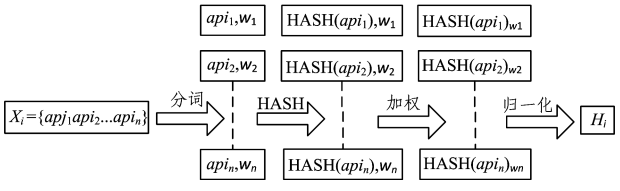


图 2　simhash 算法的处理过程

Fig. 2　Process of simhash algorithm

首先对调用序列 X_i 进行分词，并确定每个函数的权重，若调用的函数可能导致系统出现安全问题，则权重 w_k 为 2，否则 w_k 为 1；然后对每个函数都做 b-bits 的 hash 计算（图 2 中 $b=6$）和加权，若 hash 位数为 1，则 w_k 为正，否则为负；最后将加权后的权重累加和进行归一化处理，得到最终的 simhash 值 H_i，H_i 即为 API 调用序列的特征。

3.2　基于最大特征距离的样本选择策略

基于最大特征距离的样本选择策略的目的是从大量未标记样本中选择具有标记价值的样本，这一方面能够减少后续标记的工作量，提高整个主动学习算法的检测速度；另一方面能够提高标记的准确率，降低无用样本带来的干扰。

在主动学习初期，已标记的样本较少，训练的分类器泛化性能较低，难以对特征相近的样本进行预测，选择特征差异较大的样本能够降低预测难度。而同一类样本的特征存在相似性，相似性越低的样本，其特征之间的差异越大。为此，本文提出了一种基于最大特征距离的样本选择策略，将特征间的海明距离（Hamming Distance）[12]作为样本差异性的衡量标准，其目的是从未标记样本集 U 中选择出待标记的样本集 S，以为后续的样本标记提供支持。

由于特征的数据为二进制形式，因此使用海明距离能够更好地反映各特征在位数上的差异。海明距离是两个 b 位长码字，例如 $z=\{z_1 z_2 \cdots z_r \cdots z_b\}$ 和 $y=\{y_1 y_2 \cdots y_r \cdots y_b\}$ 之间对应位的不同比特总数，其公式如下：

$$D_{\mathrm{Ham}}(y,z)=\sum_{r=1}^{m} y_r \oplus z_r \tag{1}$$

其中，$y_r \in \{0,1\}$，$z_r \in \{0,1\}$，$D_{\mathrm{Ham}}(y,z)$表示 y 和 z 中在相同位置上不同比特数的总数，总数越多相似度越低。如对于两个 API 调用序列的特征 $H_1=10011011$ 和 $H_2=10101001$，其中不同的位数共有 3 位，因此 $D_{\mathrm{Ham}}(H_1,H_2)=3$。

本文假设恶意样本由于其恶意性会大量调用敏感函数，使得其 API 调用较为相似，利用该策略进行选择的过程如图 3 所示。首先对未标记样本特征集 T_U 进行两两计算，得到海明距离，并将其保存在数组中；其次计算数组元素的最大值，并返回最大距离特征指向的样本；最后选择具有最大值的两个样本加入待标记样本集 S 中。基于最大特征距离的样本选择策略的算法描述如算法 1 所示。

算法 1 基于最大特征距离的样本选择策略

输入：未标记样本集 U，共 u 个样本

输出：待标记样本集 S

Step 1 设 i=1，j=i+1，i，j∈U。

Step 2 计算第 i 个和第 j 个未标记样本的海明距离 D_{Ham}(i，j)，并将计算结果保留在集合 D 中。

Step 3 If j≤u j=j+1 and go to Step 2；
else go to Step 4。

Step 4 If i≤u−1 i=i+1 and go to Step 2；
else go to Step 5。

Step 5 计算 MAX{D}，选择符合条件的两个样本加入待标记样本集 S 中。

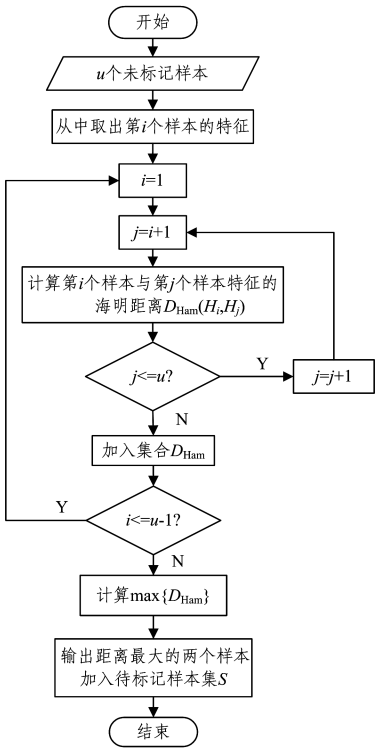


图 3 基于最大特征距离的样本选择策略流程图

Fig. 3 Flowchart of sample selection strategy based on maximum distance

3.3 基于最小估计风险的样本标记策略

基于最小估计风险的样本标记策略的目的是通过分析样本自身的信息和规律，利用机器自动对样本进行标记，以进一步提升算法的检测速度。

其基本思路是：首先将选择出的待标记样本集 S 中的样本与已标记样本集 L 中的样本进行相似性度量；然后利用初始分类器 C 对 S 中的样本进行预测；最后根据相似性度量的结果和预测的结果计算样本的估计风险值，并输出估计风险值最小的样本及其标记。在传统的主动学习过程中，从未标记样本集 U 中选择出待标记样本后通常都是采用人工交互的方式进行标记[13]，会消耗大量的人力和物力，还存在标记速度过慢的缺点，难以快速进行检测。因此，本文结合相似性度量，利用最小估计风险的方法对样本自动进行标记，该策略的流程如图 4 所示。

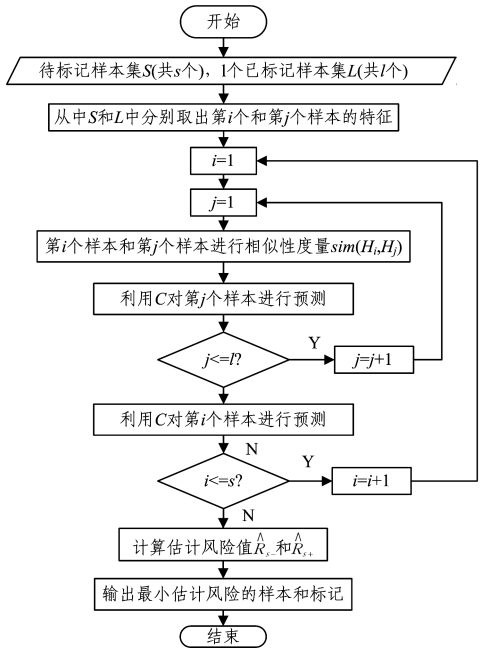


图 4 基于最小估计风险的样本标记策略的流程图

Fig. 4 Flowchart of sample tagging strategy based on Minimum Risk Estimate

在 3.2 节的海明距离计算的基础上设计相似性度量的标准，对 S 中每个样本的特征与 L 中每个样本的特征进行度量，其思想是：特征越相似的两个样本，其属于同一类样本的可能性就越大。通过与不同类样本进行相似性度量，可初步确定待选择样本属于某一类的概率。相似性度量的计算过程如下。

设来自 S 和 L 的样本特征分别为 $H_s=\{y_1 y_2 \cdots y_r \cdots y_b\}$，$H_l=\{z_1 z_2 \cdots z_r \cdots z_b\}$，定义相似性度量公式为：

$$sim(H_s,H_l)=1-(\sum_{r=1}^{m} y_r \oplus z_r)/n \tag{2}$$

其中，y_r，z_r 分别表示两个样本的特征 H_s，H_l 所对应的比特数值，H_s 表示 S 中第 s 个样本的特征，H_l 表示 L 中第 l 个样本的特征。若 T_S 与 T_L 中被标记为恶意代码的样本进行相似性度量，则用 $sim(H_s,H_{l+})$ 表示；若 T_S 与 T_L 中被标记为正常代码的样本进行相似性度量，则用 $sim(H_s,H_{l-})$ 表示，H_{l+} 表示 L 中恶意代码类样本的第 l 个样本的特征，H_{l-} 表示 L 中正常代码类样本的第 l 个样本的特征。

估计风险计算的基本思想是：对某样本的分类器 C 的预测结果与相似性度量结果进行综合评估，估计风险越低说明该样本标签被标记正确的概率就越大。该方法最早由 Zhu 等[14]提出，在此基础上，本文结合文献[8]对其进行一定的改进，结合样本相似性度量的结果和分类器 C 的预测结果进行计算，其主要过程如下：首先设 C 的预测结果为$C(H)(C(H)\in\{0,1\})$，预测的标签为 $h()(h\in\{$恶意代码，正常代码$\})$。为进一步确定样本的标记情况，在相似性度量的基础上，对 C 的预测结果与相似性度量的结果进行风险值的估计，将估计风险值最小的样本及其标记输出。这里包括对 L 中恶意代码类和正常代码类的风险估计，其计算公式如下：

$$\hat{R}_{s+}=\sum_{l+}\frac{(C(H_s)-C(H_{l+}))^2}{sim(H_s,H_{l+})} \tag{3}$$

$$\hat{R}_{s-}=\sum_{l-}\frac{(C(H_s)-C(H_{l-}))^2}{sim(H_s,H_{l-})} \tag{4}$$

其中，$\hat{R}_{s+}$ 和 $\hat{R}_{s-}$ 分别表示正类和负类的风险估计值，s 表示待标记样本集合 S 中的某一样本，$l+$ 和 $l-$ 分别表示已标记样本集合 L 中正类和负类的某一样本，$(C(H_s)-C(H_{l-}))^2$ 表示两个样本与相应类样本预测的差异。

当两个样本的相似性较高但预测差异较大时，估计的风险值较大。当样本预测结果相似或其本身相似性较低时，不会引起估计风险的增加。最后比较风险值的大小，选择风险值最低的样本及其标记输出。例如，对于待标记样本 s_1，s_2，s_3，其估计风险值最低的为$\hat{R}_{s_1+}$，则标记 $h(s_1)$ 为恶意代码。基于最小估计风险的样本标记策略的算法描述如算法 2 所示。

算法 2　基于最小估计风险的样本标记策略

输入：已标记样本集 L(共 l 个样本)，待标记样本集 S(共 s 个样本)，分类器 C

输出：标记后的样本及其标签 h()

Step 1　设 i=1，j=1，i∈S，j∈L。

Step 2　计算相似性 sim(H_i，H_j)。

Step 3　利用分类器 C 预测(C(H_i))。

Step 4　If i≤s i=i+1 and go to Step 2；
　　else go to Step5。

Step 5　If j≤l j=j+1 and go to Step 2；
　　else go to Step6。

Step 6　计算估计风险值$\hat{R}_{s+}$，$\hat{R}_{s-}$。

Step 7　输出最小估计风险值的样本及其标签 h()。

4　算法设计与实现

在第 3 节的基础上，本节设计并实现主动学习算法。该算法将已标记样本和未标记样本组成的原始样本集作为输入，实现对未标记样本的标记，其基本流程如图 5 所示。

该算法的具体描述如算法 3 所示。

算法 3　改进主动学习的恶意代码检测算法

输入：原始样本集

输出：U 中的样本标记

Step1　首先根据原始样本集的标记情况将其分为已标记样本集 L 和未标记样本集 U，然后按照第 3 节和第 4 节的方法对原始样本集进行特征提取和降维，其特征分别表示为已标记样本特征集 T_L 和未标记样本特征集 T_U。

Repeat：

Step2　利用随机森林算法对 L 进行训练，得到初始分类器 C。

Step3　利用式(1)计算 U 中两两样本间的海明距离，选择距离最大的样本加入待标记样本集 S。

Step4　利用式(2)对 S 和 L 中的各样本进行相似性度量。

Step5　利用 C 对 S 中的每个样本进行预测，得到预测结果 C()和标记 h()。

Step6　利用式(3)和式(4)计算样本的估计风险值，并选择估计风险值最小的样本及其标记加入 L。

Step7　更新 L 和 U。

Until U 中所有样本被标记完毕。

Step8　输出 U 中每个样本的检测结果和标记。

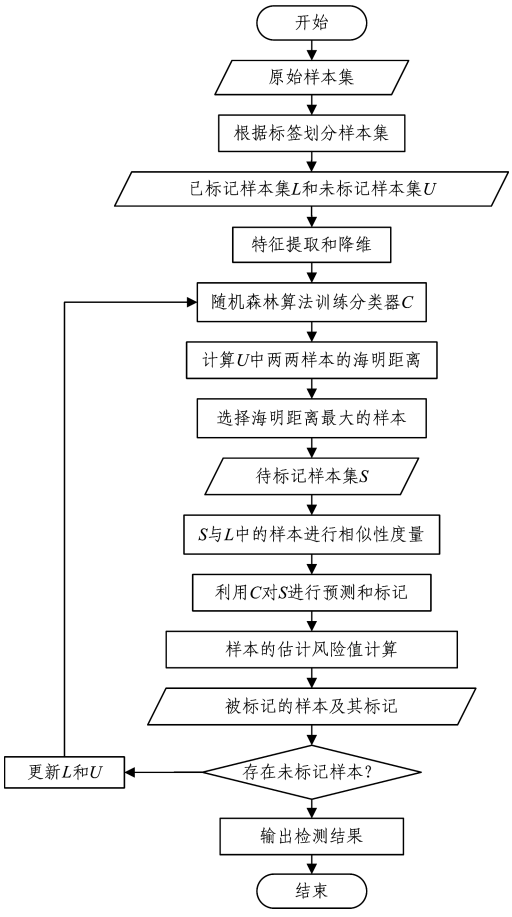


图 5　改进主动学习的恶意代码检测算法的流程图

Fig. 5　Flowchart of malware detection algorithm for improving active learning

5　实验验证

本文实验的目的是在少量已标记的恶意代码样本条件下，利用提出的主动学习算法对样本特征进行训练，提高对未标记样本的检测率。实验数据取自 VXHeavens 和 Malshare 恶意代码共享网站，共收集 Backdoor，Trojan，Virus，Worm 4 种主要类型的恶意代码，共计2949个恶意代码样本。正常样本是由.dll，.exe 系统文件或者常用的应用程序组成的，共计

2304 个。样本的分析是通过 virustotal 网站的在线沙盒分析系统实现的，在获得 API 接口后，编写 python 脚本，逐一对样本进行分析，然后提取各样本所调用的 API 函数。实验的环境配置如表 1 所列。

表 1 实验环境的设置

Table 1 Setting of experimental environment

实验主机	处理器：Inter Core i5-4690 CPU@3.5GHz 内存：16GB 磁盘空间：1TB 操作系统：window 7 64-bit
软件配置	Python2.7.9，PyCharm x64，virustotal

5.1 实验设计

本文共设计了 3 组实验。

(1)不同学习方法对检测结果的影响。共选择了 3 种学习方法来与本文的主动学习算法(MDMRE)进行对比，分别是：使用随机选择策略的主动学习，即基于最小化估计风险的主动学习(Minimum Risk Estimate，MRE)；使用人工标记策略的主动学习，即基于最大距离和人工标记的主动学习(Maximum Distance and Manual Marking，MDMM)；不使用主动学习而直接进行分类训练的方法，即普通的机器学习算法(Baseline)。

(2)样本数量对检测结果的影响。分别设置样本数量为样本总数的 10%，20%，40%，60%，80%进行实验，以评价在少量样本的情况下使用主动学习的优势与不足。

(3)已标记样本的占比对检测结果的影响。设已标记样本的占比为 $L/(L+U)$，将其分别设置为 10%，20%，40%，60%，80%进行实验，以分析 L 和 U 的比例对检测结果的影响。

实验采用正确率(Correct Rate，CR)、真阳性率(TP Rate，TPR)和假阳性率(FP Rate，FPR)作为评价指标，下面给出其含义及计算方式。

FP(False Positives)：被模型预测为正的负样本。

TP(True Positives)：被模型预测为正的正样本。

TN(True Negatives)：被模型预测为负的负样本。

FN(False Negatives)：被模型预测为负的正样本。

CR：正确分类的个数占总数的比例。计算公式为 $CR=\dfrac{TP+TN}{TP+TN+FN+FP}$。

TPR：实际正类中被预测为正类的比例。数值越大，说明预测正类的准确度高，不容易将正类预测为负类。计算公式为 $TPR=\dfrac{TP}{TP+FN}$。

FPR：实际负类中被预测为正类的比例。数值越大，说明分类器越容易将负类预测为正类，分类器效果越差。计算公式为 $FPR=\dfrac{FP}{FP+TN}$。

在实验过程中，为保证随机性，每组实验分别进行 10 次，最终的结果为 10 次实验的平均值。

5.2 实验步骤

Step1 根据实验设计，设置实验样本数量以及 L 和 U 的数量。

Step2 编写 python 脚本，将样本逐一投入 virustotal 在线沙盒系统以获取 API 调用函数[15]；然后根据函数的先后顺序将其表示成序列；最后通过 3.1 节的特征处理算法，得到各样本 simhash 后的特征。

Step3 根据实验设计，设置各组实验的样本数据。

Step4 使用 python 编写本文提出的主动学习算法和 5.3.1 节中的其他学习算法，并利用各算法对特征进行训练，根据结果计算各算法的 CR，TPR 和 FPR。

Step5 设置训练样本比例，采用本节提出的主动学习算法与 Baseline 法对不同比例下的样本进行训练，并根据结果计算 CR，TPR 和 FPR。

Step6 设置已标记样本和未标记样本的比例，采用本节提出的主动学习算法与 Baseline 法对不同 $L/(L+U)$ 的比例进行训练，并根据结果计算 CR，TPR 和 FPR。

Step7 绘制图形，评估各组实验的检测结果。

5.3 实验结果分析

表 2－表 6 分别对比了样本数量占样本总数的 10%，20%，40%，60%，80%时，4 种训练方法的 CR，TPR 和 FPR 值，同时还分别比较了已标记样本占比对检测结果的影响。

表 2 数量占样本总数的 10%时的测试结果

Table 2 Test results for each group under 10% samples

算法	$L/(L+U)=10\%$			$L/(L+U)=20\%$			$L/(L+U)=40\%$			$L/(L+U)=60\%$			$L/(L+U)=80\%$		
	CR	TPR	FPR	CR	TPR	FPR	CR	TPR	FPR	CR	TPR	FPR	CR	TPR	FPR
RG-MRE	0.8216	0.7984	0.1379	0.8441	0.8241	0.1269	0.8527	0.8293	0.1291	0.8531	0.8434	0.1211	0.8625	0.8275	0.1250
MRE	0.8072	0.7777	0.1554	0.8287	0.8088	0.1386	0.8404	0.8323	0.1339	0.8487	0.8305	0.1264	0.8543	0.8275	0.1250
MDMRE	0.8144	0.7782	0.1494	0.8342	0.8108	0.1285	0.8409	0.8187	0.1313	0.8487	0.8333	0.1294	0.8543	0.8275	0.1250
Baseline	0.7783	0.7286	0.1818	0.8187	0.8240	0.1643	0.8311	0.8235	0.1594	0.8439	0.8319	0.1395	0.8543	0.8474	0.1363

表 3 数量占样本总数的 20%时的测试结果

Table 3 Test results for each group under 20% samples

算法	$L/(L+U)=10\%$			$L/(L+U)=20\%$			$L/(L+U)=40\%$			$L/(L+U)=60\%$			$L/(L+U)=80\%$		
	CR	TPR	FPR	CR	TPR	FPR	CR	TPR	FPR	CR	TPR	FPR	CR	TPR	FPR
RG-MRE	0.8681	0.8565	0.1265	0.8723	0.8698	0.1192	0.8794	0.8718	0.1164	0.8826	0.8749	0.1096	0.8965	0.8871	0.1033
MRE	0.8494	0.8461	0.1445	0.8565	0.8494	0.1259	0.8606	0.8571	0.1218	0.8684	0.8512	0.1184	0.8798	0.8714	0.1136
MDMRE	0.8511	0.8429	0.1393	0.8587	0.8512	0.1245	0.8614	0.8522	0.1246	0.8684	0.8512	0.1184	0.8798	0.8714	0.1136
Baseline	0.8202	0.8132	0.1711	0.8324	0.8178	0.1622	0.8496	0.8423	0.1407	0.8602	0.8497	0.1379	0.8716	0.8663	0.1293

表 4　样本数量占样本总数的 40%时的测试结果

Table 4　Test results for each group under 40% samples

算法	$L/(L+U)=10\%$			$L/(L+U)=20\%$			$L/(L+U)=40\%$			$L/(L+U)=60\%$			$L/(L+U)=80\%$		
	CR	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*
RG-MRE	0.8761	0.8651	0.1179	0.8819	0.8746	0.1108	0.8907	0.8794	0.1075	0.9038	0.8841	0.1024	0.9117	0.8898	0.0971
MRE	0.8634	0.8423	0.1341	0.8731	0.8527	0.1276	0.8832	0.8611	0.1207	0.8956	0.8643	0.1159	0.9056	0.8713	0.1064
MDMRE	0.8687	0.8442	0.1326	0.8753	0.8442	0.1237	0.8851	0.8516	0.1224	0.8962	0.8617	0.1184	0.9062	0.8756	0.1079
Baseline	0.8351	0.8205	0.1562	0.8523	0.8475	0.1364	0.8611	0.8498	0.1306	0.8697	0.8544	0.1284	0.8807	0.8713	0.1194

表 5　样本数量占样本总数的 60%时的测试结果

Table 5　Test results for each group under 60% samples

算法	$L/(L+U)=10\%$			$L/(L+U)=20\%$			$L/(L+U)=40\%$			$L/(L+U)=60\%$			$L/(L+U)=80\%$		
	CR	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*
RG-MRE	0.8873	0.8545	0.1141	0.8995	0.8605	0.1103	0.9006	0.8879	0.1055	0.9112	0.8973	0.0995	0.9208	0.8961	0.0831
MRE	0.8703	0.8419	0.1282	0.8774	0.8486	0.125	0.8852	0.8511	0.1194	0.9012	0.8816	0.1107	0.9128	0.8879	0.0991
MDMRE	0.8787	0.8365	0.1263	0.8816	0.8446	0.1211	0.8889	0.8512	0.1176	0.9084	0.8902	0.1110	0.9173	0.8842	0.0974
Baseline	0.8562	0.8231	0.1419	0.8628	0.8361	0.1355	0.8762	0.8471	0.1309	0.8819	0.8747	0.1243	0.9013	0.8641	0.1085

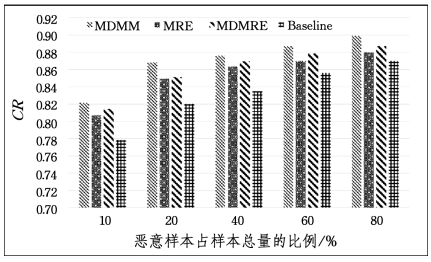
表 6　样本数量占样本总数的 80%时的测试结果

Table 6　Test results for each group under 80% samples

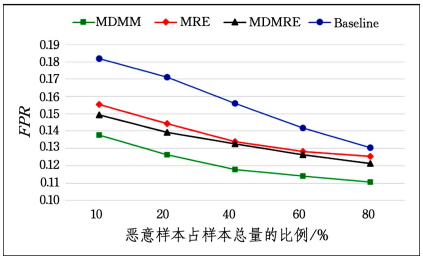
算法	$L/(L+U)=10\%$			$L/(L+U)=20\%$			$L/(L+U)=40\%$			$L/(L+U)=60\%$			$L/(L+U)=80\%$		
	CR	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*	*CR*	*TPR*	*FPR*
RG-MRE	0.8994	0.8832	0.1105	0.9005	0.8944	0.1077	0.9258	0.8999	0.0906	0.9189	0.9043	0.0812	0.9326	0.9154	0.0687
MRE	0.8796	0.8643	0.1256	0.8914	0.8843	0.1195	0.9035	0.8973	0.0934	0.9127	0.8974	0.0888	0.9215	0.9096	0.0707
MDMRE	0.8870	0.8616	0.1213	0.8967	0.8742	0.1153	0.9092	0.8943	0.0962	0.9145	0.9017	0.0861	0.9253	0.9136	0.0711
Baseline	0.8704	0.8661	0.1304	0.8789	0.8663	0.1264	0.8922	0.8841	0.1176	0.8984	0.8844	0.0933	0.9188	0.8927	0.0892

5.3.1　不同的学习方法对检测结果的影响

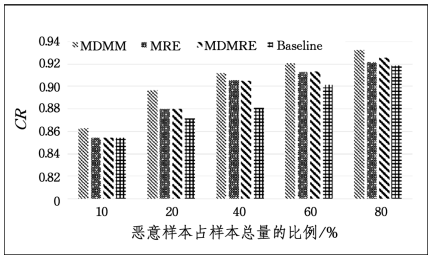
为横向比较本文算法的优势与不足,在 $L/(L+U)$ 为 10%和 80%的环境下,对不同的主动学习策略在不同样本比例下的恶意代码检测影响分别做了实验,结果如图 6 所示。根据图 6(a)和图 6(c),采用主动学习的方法和未采用主动学习的 Baseline 方法的检测正确率都随着学习样本数量的增加而提升。其中,使用 Baseline 方法的检测正确率均明显低于其他方法,这说明使用主动学习方法进行分类训练的效果优于直接进行分类训练的效果。此外,由图 6(b)和图 6(d)可知,使用主动学习方法的分类器效果也明显优于 Baseline 方法的分类效果,从而进一步说明了主动学习方法的优越性。



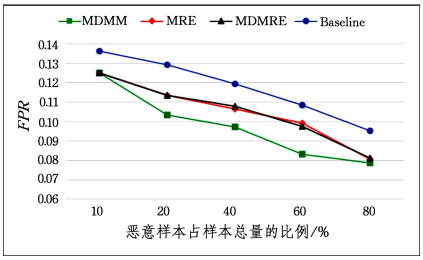
(a)CR ($L/(L+U)=10\%$)



(b) FPR ($L/(L+U)=10\%$)



(c)CR ($L/(L+U)=10\%$)



(d) FPR ($L/(L+U)=10\%$)

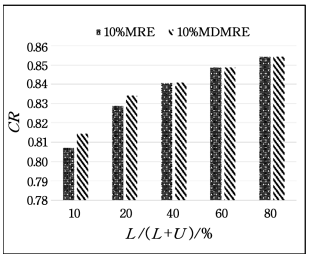
图 6　不同学习方法下的检测结果

Fig. 6　Test results of different learning methods

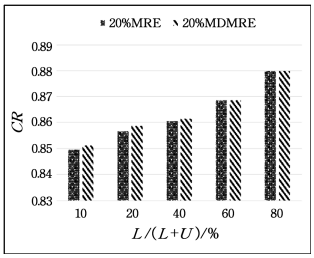
不同学习策略对检测结果的影响也不同。由图 6(a)和图 6(c)可知,采用人工进行标记的 MDMM 方法的检测正确率显著高于其他算法的正确率,这主要是因为人工标记的正确率很高,但其花费的时间和人力成本也很高。因此,在具有足够的时间、人力和物力的条件下,基于 MDMM 的主动学习方法是最优选择。而使用 MRE 的方法与本文提出的 MDM-

RE 方法相比，在已标记样本数量占比为 10%的情况下，MDMRE 的检测正确率高于 MRE 方法；但是在图 6(c)中，在已标记样本数量占比为 80%的情况下，MDMRE 的检测正确率与 MRE 方法的正确率几乎相同。可以推测，在未标记样本数量占比较大的情况下，使用 MDMRE 方法较好。

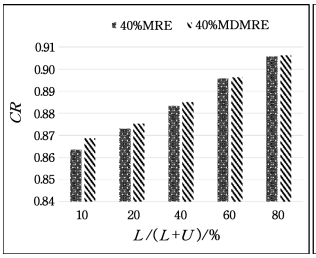
为进一步对推测进行验证，分别将 $L/(L+U)$设置为 10%，20%，40%，60%，80%进行实验，并对不同比例的样本进行对比，实验结果如图 7 所示。由图 7(a)－图 7(c)可以看出，在 $L/(L+U)$为 40%以后，MDMRE 方法和 MRE 方法的准确率相差不大，几乎持平；而其他情况下，MDMRE 方法均优于 MRE 方法。这可能是由于未标记样本的数量较少，采用随机选择法与最大特征距离选择法每次选择相同样本的概率较大，而标记策略都是最小估计风险法，因此样本和标记的结果都相同，从而使检测的结果相差不大。对于样本数量较大的样本集，相同的 $L/(L+U)$比例下，未标记样本的数量仍然较大，因此两种选择策略难以选择相同的样本，从而使得基于最大距离选择策略的主动学习更有优势。



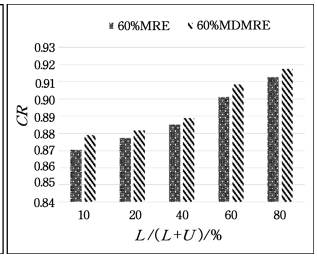
(a)CR (10%($L+U$))



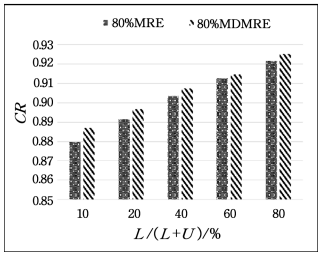
(b)CR (20%($L+U$))



(c)CR (40%($L+U$))



(d)CR (60%($L+U$))



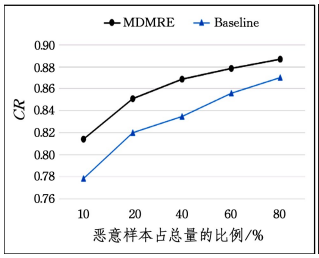
(e)CR (80%($L+U$))

图 7 MRE 和 MDMRE 在不同未标记样本占比情况下的检测结果

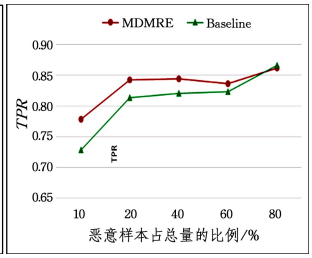
Fig. 7 Detection results of MRE and MDMRE in proportion of different unlabeled samples

5.3.2 不同样本数量大小对检测结果的影响

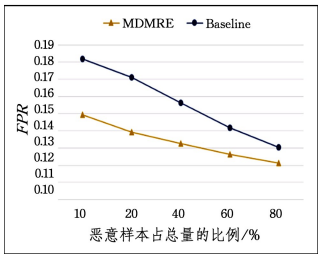
本文提出的基于 MDMRE 主动学习算法的目的是通过少量样本来提高对恶意代码样本的检测率，因此用于训练的样本量的大小也是影响检测结果的因素。为了验证主动学习在少量样本下的检测效果。对比了 MDMRE 方法和 Baseline 方法在不同样本数量的检测效果，为保证实验的准确性，控制变量 $L/(L+U)$的比例为 10%，实验结果如图 8 所示。可以看出，MDMRE 法在少量样本下的检测效果优于 Baseline 法。



(a)CR



(b)TPR



(c)FPR

图 8 MDMRE 和 Baseline 在不同学习样本数量下的检测结果

Fig. 8 Detection results of MDMRE and Baseline under different number of learning samples

如图 8(a)所示，随着学习样本数量的逐渐增多，两种方法的检测正确率不断提升。在样本较少的阶段，MDMRE 的检测正确率明显高于 Baseline 的检测准确率，但随着样本数量的增加，两种方法的检测正确率的差值越来越小。在图 8(b)中，随着样本的增加，预测正类样本的正确率提升，但在 40%之后，两种方法的真阳性率的差值逐渐缩小，在 80%处几乎相等。在图 8(c)中，随着训练样本数量的增加，分类器的性能增加，因此假阳性率的曲线逐渐走低；此外，两种方法的假阳性率的差值也逐渐缩小。造成上述现象的原因可能是，当未知样本数量较大时，标记的难度逐渐增加，从而导致正确率降低。由上述分析可得，MDMRE 法在少样本下的检测效果优于 Baseline 法，但随着样本量的增加，其检测效果的优势逐渐减少。

5.3.3 标记样本的占比对检测结果的影响

在主动学习过程中，已标记样本的数量对分类器的效果具有一定影响，已标记样本越多，其训练的初始分类器性能越好。而未标记样本越多，样本选择和标记的难度就越大。可见，标记样本和未标记样本数量的比例也是影响检测结果的因素之一。本文对提出的 MDMRE 主动学习方法和 Baseline 法进行实验，结果如图 9 所示，横坐标表示用于训练的总样本量，纵坐标表示 MDMRE 与 Baseline 检测正确率的差值，即 $|CR(\text{MDMRE})-CR(\text{Baseline})|$。

如图 9 所示，训练样本占样本总数的 10%时，$L/(L+U)$为 10%的差值显著高于其他几个比例下的差值，这说明在这种比例下 MDMRE 法的检测正确率较高于 Baseline。而随着训练样本数量的增加，10%的差值不再显著，逐渐与其他比例

下的差值趋平。在 40%($L+U$)的情况下，10%的差值远小于 40%的差值，可能是由于已标记样本中恶意代码和正常样本的比例失衡，导致 Baseline 法训练的分类器效果很差，而 MDMRE 法不断增加标记的样本，使得比例得以平衡，然后重新训练分类器，使得分类器的性能逐步提高，因此两种方法的检测准确率的差值进一步增大。根据上述分析，在 $L/(L+U)$ 值较小的情况下，使用基于 MDMRE 主动学习的恶意代码检测算法的效果更好。

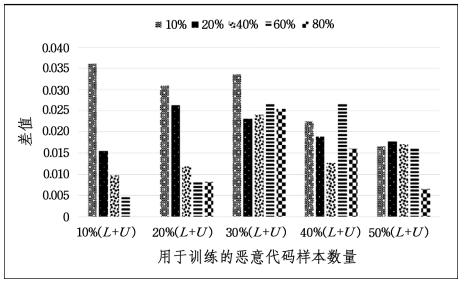


图 9　MDMRE 和 Baseline 在不同比例下的检测正确率的差值

Fig. 9　Differences in detection accuracy of MDMRE and Baseline at different ratios

结束语　本文针对恶意代码出现之初，用于分析检测的样本数量较少的现实情况，在原有的主动学习算法的基础上设计了基于最大距离的样本选择策略和最小估计风险的样本标记策略，并通过实验验证了该算法的可行性。实验结果显示，在 $L/(L+U)$ 较小的情况下，采用该算法比采用随机选择策略的主动学习算法的检测效果更优。在时间性能上，采用该算法比采用人工标记策略的主动学习算法更具优势。同时，与不采用主动学习的机器学习算法相比，该算法具有更好的检测效果，而且在少量样本集和 $L/(L+U)$ 较小的情况下，该算法的优势更加明显。可见，本文提出的算法具有一定的实用价值，能够为当前新型恶意代码的检测提供有效的技术支持。

但是，实验发现所提算法的标记策略的正确率与人工标记的正确率还存在一定的差距，而且其选择策略在未标记样本较少的情况下的检测效果较差。未来将改进选择策略，对样本标记的难度值进行排序，并采用人工标记的方式对难度大的样本进行标记，以进一步改善检测效果。

参 考 文 献

[1] LIU J, SU P R, YANG M, et al. Software and Cyber Security-A Survey [J]. Journal of Software, 2018, 29(1): 42-68. (in Chinese)
刘剑，苏璞睿，杨珉，等. 软件与网络安全研究综述[J]. 软件学报，2018，29(1)：42-68.

[2] TONG S, CHANG E. Support vector machine active learning for image retrieval[C]// Proceedings of the 9th ACM International Conference on Multimedia. New York: ACM, 2001: 107-118.

[3] TONG S, KOLLER D. Support vector machine active learning with applications to text classiflcation[J]. The Journal of Machine Learning Research, 2002, 2(1): 999-1006.

[4] CHEN Y D, WANG T, CHEN H W. Combining Semi-Supervised Learning and Active Learning for Shallow Semantic Parsing[J]. Journal of Chinese Information Processing, 2008, 22(2): 70-75. (in Chinese)
陈耀东，王挺，陈火旺. 半监督学习和主动学习相结合的浅层语义分析[J]. 中文信息学报，2008，22(2)：70-75.

[5] JOACHIMS T. Transductive Inference for Text Classification using Support Vector Machines[C]// Sixteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., 1999: 200-209.

[6] SEUNG H S, OPPER M, SOMPOLINSKY H. Query By Committee[C]// Proceedings of the 15th Annual ACM Workshop on Computational Learning Theory. California: ACM, 1992: 287-294.

[7] FREUND Y, SEUNG H S, SAMIR E, et al. Selective Sampling Using the Query By Committee Algorithm[J]. Machine Learning, 1997, 28(23): 133-168.

[8] MAO W X, CAI Z M, TONG L. Malware Detection Method Based on Active Learning [J]. Journal of Software, 2017, 28(2): 384-397. (in Chinese)
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.

[9] MANKU G S, JAIN A, SARMA A D. Detecting near-duplicates for web crawling[C]// Proceeding of the 16th International Conference on World Wide Web. USA: ACM Press, 2007: 141-149.

[10] ZHENG Y, WANG Y J, XUE Z. Android Malware Detection of Calls Tracing with Android Manifest and API[J]. Journal of Computer Research and Development, 2017(3): 126-130. (in Chinese)
郑尧，王轶骏，薛质. 通过 Android Manifest 和 API 调用追踪的恶意检测[J]. 计算机技术与发展，2017(3)：126-130.

[11] DUAN X Y. Research on the Malware Detection Based on Windows API Call Behavior[D]. Chengdu: Southwest Jiaotong University, 2016. (in Chinese)
段晓云. 基于 Windows API 调用行为的恶意软件检测研究[D]. 成都：西南交通大学，2016.

[12] ZHANG H J. Text Similarity Computing Based on Hamming Distance[J]. Computer Engineering and Applications, 2001, 37(19): 21-22. (in Chinese)
张焕炯. 基于汉明距离的文本相似度计算[J]. 计算机工程与应用，2001，37(19)：21-22.

[13] LIU D Y, QIU W J. Active Learning for Multi-label Classification Based on SVM's Expect Margin[J]. Computer Science, 2011, 38(4): 230-232. (in Chinese)
刘端阳，邱卫杰. 基于 SVM 期望间隔的多标签分类的主动学习[J]. 计算机科学，2011，38(4)：230-232.

[14] GOKHAN T, DILEK H, ROBERT E. Combining active and semi-supervised learning for spoken language understanding[J]. Speech Communication, 2005, 45(2): 171-186.

[15] LI Z Y. A Automatic Detection Method of Malware Behavior Based on Sandbox[D]. Wuhan: Huazhong University of Science and Technology, 2015. (in Chinese)
李志勇. 基于沙箱技术的恶意代码行为自动化检测方法[D]. 武汉：华中科技大学，2015.