

基于属性序约简的恶意代码检测

郭宁¹, 孙晓妍², 林和¹, 牟华³

(1. 兰州大学 信息科学与工程学院, 兰州 730000; 2. 信息工程大学 信息工程学院, 郑州 450002; 3. 山东省情报研究所, 济南 250000)
(mr_dingding@163.com; iamsxy666@sina.com)

摘要:研究了已有的恶意代码特征选择和约简方法, 针对已有的属性约简方法没有充分利用特征选择评估函数信息的不足, 提出以信息增益值和特征的规模对候选特征排序, 并使用属性序约简对特征进行约简的方法, 分析了时空复杂度, 给出了总体设计方案。实验结果验证了属性序约简的应用能够在较短的时间内获得较少的约简结果, 使用约简后的特征进行分类准确率较高。

关键词:恶意代码; 特征选择; 约简; 信息增益; 属性序

中图分类号: TP309.5 **文献标志码:** A

Malware detection based on attributes order reduction

GUO Ning¹, SUN Xiao-yan², LIN He¹, MOU Hua³

(1. College of Information Science and Engineering, Lanzhou University, Lanzhou Gansu 730000, China;
2. College of Information Engineering, PLA Information Engineering University, Zhengzhou Henan 450002, China;
3. Information Research Institute of Shandong Province, Jinan Shandong 250000, China)

Abstract: The existing methods of malware feature selection and reduction methods were studied. Current attribute reduction methods of malware do not take advantage of the information of feature selection evaluation function. So a method was proposed to order all features based on their value of information gain and their size, and used attributes order reduction method to get a reduction. An analysis of spatial and temporal complexity was given, and the overall design was given. Test results show that the application of attributes order reduction can obtain fewer reduction results in less time, and get higher classification accuracy using the reduction result.

Key words: malware; feature selection; reduction; information gain; attribute order

受各种利益的驱使, 恶意代码的数量仍然在大幅增长, 安天实验室信息安全威胁综合报告^[1]中指出, 2010年上半年, 安天实验室捕获恶意代码样本4447713个, 与2009年同期相比增长率为167.2%, 其中木马、蠕虫、后门及其他类恶意代码增长量较大, 攻击的目的包括窃取信息、破坏大型基础设施等。例如, 恶意代码“震网(Stuxnet)”^[2]对工业基础设施进行破坏, 带来了巨大的威胁。

面对如此大量的恶意代码, 仅仅依靠安全人员的逆向工程或者其他手工方法进行分析, 势必造成巨大的工作量。将恶意代码分析向智能化、自动化发展已成为当前的发展趋势。本文研究基于数据挖掘和机器学习的恶意代码的智能分析, 重点是研究粗糙集约简在恶意代码智能检测中的应用。

1 研究现状

在基于数据挖掘和机器学习的恶意代码检测上, 主要分为特征提取、特征选择和分类三个过程^[3]。

根据针对的提取方式的不同, 特征提取可以分为静态提取和动态提取。静态提取主要针对的是静态二进制文件和经过反汇编得到的汇编指令, 提取的特征可以是n_gram字节序列、PE头信息、导入导出信息、资源目录、版本信息、出现的字符串、n_gram指令序列以及指令使用频率等。动态提取主要是在一个虚拟或仿真环境下动态执行恶意代码^[4-5], 获取恶

意代码的执行行为, 主要指函数调用, 提取的特征可以是系统调用频率和n_gram调用序列。静态提取快速、全面, 但受代码迷惑技术的影响; 动态提取虽然耗时, 但不受代码迷惑技术影响。动态提取已成为当前的研究热点, 但是动态提取的特征主要针对调用名称和调用序列, 很少考虑调用所针对的对象。

特征选择上, 在文本特征选择上使用的选择方法^[6]在恶意代码特征选择上得到了应用, 例如信息增益^[7]、增益比、互信息^[8]和文档频率等。经过上述选择可能存在一些冗余的特征, 因此, 文献^[9]使用粗糙集属性约简来去除冗余的属性, 进一步地降低属性规模, 取得了较好的结果。但是, 它针对的特征是n_gram字节序列, 容易受代码迷惑的影响, 同时使用的属性约简为基于正区域的属性约简方法, 并没有使用特征选择前期的有用信息。通过研究发现, 可以使用在特征选择前期的信息来指导属性约简, 进而加快属性约简的过程并得到数目较少的约简结果。

在分类方法上, 主要使用支持向量机、神经网络、决策树、朴素贝叶斯等方法, 由于不是本文的研究重点, 在实验环节中, 使用对小数据样本效果较好的支持向量机进行分类。

针对上述问题, 本文对动态提取的特征的选择和约简进行研究, 主要集中于属性序约简在恶意代码检测中的应用, 期望在简短的时间内得到较少的特征, 同时保持较好的分类能力。

收稿日期: 2010-10-18; 修回日期: 2010-12-01。

作者简介: 郭宁(1981-), 男, 山西太原人, 硕士研究生, 主要研究方向: 数据挖掘; 孙晓妍(1980-), 女, 山东威海人, 博士研究生, 主要研究方向: 信息安全; 林和(1963-), 男, 甘肃临洮人, 副教授, 主要研究方向: 人工智能、数据挖掘; 牟华(1979-), 女, 山东济南人, 工程师, 主要研究方向: 数据挖掘。

2 特征提取

随着基于行为的检测的出现,针对系统调用的检测更加普遍。通过虚拟化的环境运行获取代码的执行踪迹已成为当前恶意代码分析的重要途径。这个虚拟的环境可以是运行于虚拟机软件(如VMware)、PC Emulator(如QEMU)上的操作系统。因为虚拟软件(如VMware、仿真PC QEMU)不仅能够模拟一个真实的系统,防止恶意程序对真实的系统造成危害,同时还允许保存虚拟机内部的快照,以便在虚拟机内部系统遭受恶意程序破坏以后,快速恢复到一个干净的状态。获取系统调用的方法或者是在系统内挂钩、或者是在虚拟机外通过修改仿真软件进行捕获。

接下来针对这个获取的调用行为进行特征的提取。

每个系统调用都表示对资源或服务的一个操作,每个系统调用实际上也有一个操作对象,这个对象的不同经常导致操作目的的不同,因此,可以针对操作和操作对象进行特征的提取,这时特征携带了一定的空间信息。对操作和操作对象进行特征的提取,需要将调用踪迹进行预处理,将相同行为的不同调用进行统一,将所有的新建文件用统一的文件名和后缀名标识,最后得到“操作行为”的形式:OP(O),OP是操作名,O是操作的对象,然后基于这个信息进行特征的选择。

采用系统调用的n_gram分析能够捕获一定的时序信息,例如系统调用序列为ABCDEFGH,则活动窗口为4的n_gram集为{ABCD, BCDE, CDEF, DEFG},研究表明n_gram集作为特征比调用频率作为特征在分类效果上准确率更高。

在训练样本中寻找各特征是否出现,若出现,则表示为1;否则,表示为0。最后有一个属性表示样本是否恶意,恶意样本约定恶意样本则表示为1;否则,表示为0。这样,每个样本都使用一个向量来表示。下面针对这个向量空间进行特征选择。

3 特征选择

特征选择方法在文本分类中得到了广泛的应用,进行特征选择的方法很多,依据的因素包括频度、集中度、分散度等。已有的实验结果表明信息增益是最有效的特征选择算法之一,同时文献[10]也通过定性的评估特征选择函数性能,指出信息增益是最有效的特征选择算法之一。因此,利用信息增益来选择特征。

3.1 信息增益

信息增益是一种基于熵的评估方法,它从信息论的角度出发进行特征选择。

在文本分类的特征选择中,信息增益衡量的是某个特征的出现与否对判断文本是否属于某个类别所提供的信息量,被定义为某一特征在文本中出现前后的信息熵之差。假设共有 m 个类别 $\{c_1, c_2, \dots, c_m\}$,根据上述定义,特征 t 的信息增益计算公式如下:

$$\begin{aligned} IG(t) &= H(C) - H(C|t) = \\ &= -\sum_{i=1}^m p(c_i) \lg p(c_i) + p(t) \sum_{i=1}^m p(c_i|t) \lg p(c_i|t) + \\ &= p(\bar{t}) \sum_{i=1}^m p(c_i|\bar{t}) \lg p(c_i|\bar{t}) = \\ &= p(t) \sum_{i=1}^m p(c_i|t) \lg \frac{p(c_i|t)}{p(c_i)} + \\ &= p(\bar{t}) \sum_{i=1}^m p(c_i|\bar{t}) \lg \frac{p(c_i|\bar{t})}{p(c_i)} \end{aligned}$$

其中: $p(t)$ 表示训练集中文本包含特征 t 的概率, $p(\bar{t})$ 表示训

练集中文本不包含特征 t 的概率, $p(c_i)$ 表示训练集中文本属于 c_i 类的概率, $p(c_i|t)$ 表示包含特征 t 的文本属于 c_i 类的概率, $p(c_i|\bar{t})$ 表示不包含特征 t 的文本属于 c_i 类的概率。

得到每个特征的信息增益值,选取排序在前面的特征作为候选属性。

基于信息增益的特征选择,并不能够消除冗余特征,在检测中为了加速检测的响应速度,使用越少的特征越好,因此有必要去除这些冗余的属性。

3.2 属性序约简

粗糙集(Rough Set)的基本思想是通过案例库的分类归纳出概念和规则,近年来在机器学习、数据挖掘等多个领域得到了广泛应用。粗糙集的属性约简是粗糙集的主要研究内容。

既然在前一步使用信息增益选择特征,每个特征的信息增益值对后面的约简有一定的指导作用,因此,根据信息增益的值对各个属性进行排序。每个属性就是各个特征,当各个属性大小(特征长度)差别较大时,例如,操作对象是某个很长的注册表值,有的操作对象仅为根目录,这时,基于检测快速性的考虑,应该在两者能力相同的情况下选择较短的操作行为,这是符合检测需求的。

基于这个指导信息,本文使用属性序下的约简来得到约简结果。

定义1 属性序^[4]。令 $S = \langle U, A = C \cup D, V, f \rangle$ 是一个决策表,在 C 上定义了一个完整的序关系“ $<$ ”,同时,为 C 中的所有属性分别标上1到 $|C|$ 。这样,在 C 上就得到了一个关于属性的序列,称为“属性序” $SO: c_1 < c_2 \dots < c_{|C|}$ 。

根据特征信息增益的值和特征大小给特征进行排序,信息增益大且规模小的特征排在前面,形成一个属性序: $p_1 < p_2 < \dots < p_{|P|}$ 。依据第2章的方式对属性建立决策表,决策属性为 mal , $mal = 1$ 表示恶意样本, $mal = 0$ 表示正常样本,其余的属性为条件属性。然后利用属性序约简算法对这个决策表进行属性约简。

基于分辨矩阵的属性序约简算法通常需要很大的空间。胡峰等人^[11]提出基于分治策略的属性序下的约简算法,需要空间复杂度仅为 $O(|U| + |C|)$,平均时间复杂度为 $O(|U| \times |C| \times (|C| + \lg |U|))$ 。本文使用基于分治策略的属性序下的快速约简算法。

这样,基于信息增益和属性自身大小的属性序约简过程如下。

1) 排序:将候选属性根据信息增益值和属性自身大小进行排序。

2) 约简:利用属性序下快速约简算法进行约简得到约简属性集合。

3.3 复杂度分析

下面分析基于信息增益和属性自身大小的属性序约简过程的时空复杂度。

首先分析排序过程的时空复杂度,假设有 $|C|$ 个属性,则快速排序最坏时间复杂度为 $O(|C|^2)$,平均时间复杂度为 $O(|C| \lg |C|)$,空间复杂度为 $O(|C|)$ 。

约简过程平均时间复杂度为 $O(|U| \times |C| \times (|C| + \lg |U|))$,空间复杂度为 $O(|U| + |C|)$ 。

这样,总体平均时间复杂度为 $O(|U| \times |C| \times (|C| + \lg |U|))$,空间复杂度为 $O(|U| + |C|)$ 。

虽然时间复杂度看上去并不理想,但当 $|U| \ll |C|$ 时,通常情况下非空标签属性的数目并不会很多,如对随机生成的相容决策表,只需大约 $\lg |U|$ 个属性就可以将对象区分开,因此,总体平均时间复杂度降为 $O(|C| \lg |C| + |U| \lg^2 |U|)$,因此当 $|U| \ll |C|$ 时,属性序下快速约简算法的应用效率很高。

3.4 与正区域约简比较

在粗糙集的属性约简研究领域中,基于正区域的约简方法得到了广泛的研究。

基于正区域的方法需要频繁的计算正区域,因此计算正区域的时间复杂度很大程度上影响了属性约简算法的时间复杂度。表 1 列出了当前比较高效的基于正区域的约简算法的时间,主要包括基于快速排序、基数排序^[12]和 Hash^[13]的方法。

表 1 正区域约简方法时间复杂度

方法	时间复杂度
快速排序	$O(C ^2 U \lg U)$
基数排序	$\max\{O(C U), O(C ^2 U/C)\}$
Hash 方法	$O(C ^2 U/C)$

从时间复杂度上来说,基数排序和 Hash 的方法相对较好,快速排序次之。当 $|U| \ll |C|$ 时,基于正区域的约简算法的时间复杂度最小为 $O(|C|^2 |U/C|)$ 。

而 3.3 节中指出,当 $|U| \ll |C|$ 时,通常情况下非空标签属性的数目并不会很多,总体平均时间复杂度降为 $O(|C| \lg |C| + |U| \lg^2 |U|)$,比 $O(|C|^2 |U/C|)$ 要高效。因此,在当 $|U| \ll |C|$ 时,选择使用属性序约简。

4 总体设计框架

通过对以上的研究,设计的总体框架如图 1 所示。

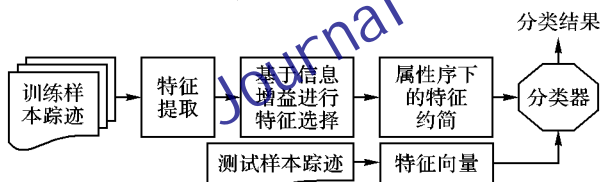


图 1 总体框架

- 1) 在虚拟环境中执行训练样本,并得到训练样本的执行踪迹。
- 2) 使用基于信息增益来得到排序在前面的作为候选特征,即进行特征选择。
- 3) 针对上述选择的特征,根据信息增益值和特征大小进行排序,使用属性序下的特征约简算法进行特征约简。
- 4) 针对选择的特征对每个训练样本建立特征向量,训练分类器。
- 5) 对测试样本,获取执行踪迹。
- 6) 针对测试样本建立特征向量。
- 7) 对测试样本的特征向量,由分类器进行分类。

5 实验验证

从初始安装的 Windows XP 系统下获取正常执行文件,从网络上收集了 152 个蠕虫和 180 个木马程序。分别将其放入虚拟机中执行,获取它们的执行踪迹。

下面使用两个实验数据集:

数据集 1 152 个蠕虫和 152 个正常文件,随机选择 76 个蠕虫和 76 个正常文件作为训练样本,剩余的样本作为测试数据。

数据集 2 180 个木马和 180 个正常文件,随机选择 90 个木马和 90 个正常文件作为训练样本,剩余的样本作为测试数据。

5.1 特征选择和约简结果

由于恶意代码会使用一些罕见的行为,因此不能使阈值设置得过高,经过实验确定特征选择的阈值为 0.003。从数据集 1 的训练样本中共获取 2 079 个操作行为,取前 1 013 个操作行为作为候选特征;从数据集 2 的训练样本中共获取 9 476 个操作行为,取前 2 555 个操作行为作为候选特征。

同样获得了两个数据集测试样本的 n_gram ($n = 4$) 短序列特征,数据集 1 的训练样本得到的短序列(大小为 4)为 3 045,取前 1 709 个短序列;数据集 2 的训练样本得到的短序列(大小为 4)为 7 738,取前 2 500 个短序列。

可以看出 $|U| \ll |C|$,使用属性序约简时间复杂度较低。通过属性序的约简分别对上述的候选特征进行约简,得到的结果如表 2。

表 2 选择和约简结果

数据集	原始数目		约简后数目	
	操作行为	API 序列	操作行为	API 序列
数据集 1	2 079	3 045	1 013	1 709
数据集 2	9 476	7 738	2 555	2 500

可以看出,约简后的特征数量明显的减少,远小于 $|C|$,与 $\lg |U|$ 接近,与 3.3 节中分析相符。在时间开销上,以已有工作使用的基于核的正区域的属性约简算法为例,得到计算核的时间消耗,进而可以判断基于正区域的属性约简的时间消耗情况,下面给出属性序约简和基于正区域的属性约简的时间开销比较,如表 3。

表 3 约简时间开销比较

属性数目	属性序约简时间/s	基于正区域约简时间/s
1 013	32	≥ 376
1 709	86	$\geq 1 110$
2 555	192	$\geq 2 819$
2 500	211	$\geq 3 066$

因此,使用基于信息增益的属性序约简明显比不使用信息增益信息而仅使用正区域约简的时间消耗要低,降低了训练时间,同时约简结果属性数目较少,达到了设计要求。

5.2 分类测试结果

下面使用上述约简的特征对数据集 1 的测试样本和数据集 2 的测试样本进行分类测试,并与没有使用属性序约简的 500 个特征(按信息增益值排序取前 500 个)进行分类的结果进行比较。使用的分类器为支持向量机,参数为默认参数。评价指标如下:

恶意程序被判定为恶意程序的比率,称为 TPR (True Positive Rate);正常程序被判定为正常程序的比率,称为 TNR (True Negative Rate);分类准确率为 $(TP + TN)/\text{总样本数}$,其中,TP (True Positive) 为恶意程序被判定为恶意程序的情况, TN (True Negative) 为正常程序被判定为正常程序的情况。

首先给出利用操作行为特征对数据集 1 的测试数据进行

分类的比较,如表4。

表4 数据集1的操作行为特征分类结果比较

操作行为特征数	TPR/%	TNR/%	准确率/%
500	96.05	90.79	93.42
8	94.74	94.74	94.74

从表4中可以看出,经过属性序约简后 TNR 和分类准确率有所提高,TPR 有所下降。

下面给出利用 API 短序列特征对数据集1的测试数据进行分类的比较,如表5。

表5 数据集1的 API 短序列特征分类结果比较

API 短序列特征数	TPR/%	TNR/%	准确率/%
500	97.37	100	98.68
6	98.68	100	99.34

从表5中可以看出,经过属性序约简后 TPR 和分类准确率得到了提高,TNR 保持不变。

下面给出利用操作行为特征对数据集2的测试数据进行分类的比较,如表6。

表6 数据集2的操作行为特征分类结果比较

操作行为特征数	TPR/%	TNR/%	准确率/%
500	95.56	85.56	90.56
10	94.44	94.44	94.44

从表6中可以看出,结果与表4基本相似,TNR 和分类准确率有所提高,TPR 有所下降。

下面给出利用 API 短序列特征对数据集2的测试数据进行分类的比较,如表7。

表7 数据集2的 API 短序列特征分类结果比较

API 短序列特征数	TPR/%	TNR/%	准确率/%
500	93.33	93.33	93.33
7	95.56	93.33	94.44

从表7中可以看出,结果与表5基本相似,TPR 和分类准确率有所提高,TNR 保持不变。

从上述结果中可以看出,利用属性序约简后的数量极少的特征进行分类的准确率比没有经过约简使用500个特征进行分类的准确率都要高,TNR 或有所提高或保持不变,而TPR 或有所提高或有所降低。虽然表4和表6的TPR 有所降低,但是分类准确率和 TNR 都有所提高,对于恶意代码检测而言,更倾向于降低误报率,即提高 TNR,因此符合检测的要求。同时,特征数目较少可以降低匹配复杂度,提高检测速度和响应能力。

6 结语

随着恶意代码数量的日益增多,智能化的代码危害性分析对恶意代码的检测和分析很有帮助。本文对基于调用踪迹的恶意代码自动分类进行了研究,重点对特征选择和约简进行研究,提出使用特征选择的信息来指导特征约简,使用属性序下的快速约简算法在较短的时间内获得较少数量的约简集。最后用实验验证了本文方法的有效性,使用约简集进行分类的效果较好。

参考文献:

- [1] 安天实验室. 安天实验室信息安全威胁综合报告[EB/OL]. [2010-07-12]. http://www.antiy.com/cn/security/2010/first_half_of_2010_the_Antiy_Laboratory_of_information_security_threats_roundup.pdf.
- [2] 安天实验室安全研究与应急处理中心. 对 Stuxnet 蠕虫攻击工业控制系统事件的综合报告[EB/OL]. [2010-09-02]. http://www.antiy.com/cn/security/2010/Report_On_the_Attacking_of_Worm_Struxnet_by_antiy_labs.htm.
- [3] SHABTAI A, MOSKOVITCH R, ELOVICI Y, *et al.* Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey [J]. Information Security Technology Report, 2009, 14(1): 16-29.
- [4] WILLIAMS C, HOLZ T, FREILING F. Toward automated dynamic malware analysis using cwsandbox [J]. IEEE Security and Privacy, 2007, 5(2): 32-39.
- [5] ANUBIS. Anubis: Analyzing unknown binaries [EB/OL]. [2010-09-16]. <http://anubis.iseclab.org>.
- [6] 刘赫. 文本分类中若干问题研究[D]. 长春: 吉林大学, 2009.
- [7] AHMED F, HAMEED H, SHAFIQ M, *et al.* Using spatio-temporal information in API calls with machine learning algorithms for malware detection and analysis [EB/OL]. [2009-08-24]. <http://nexginrc.org/nexginrcAdmin/PublicationsFiles/aiesec09-faraz.pdf>.
- [8] 陈亮, 郑宁, 郭艳华, 等. 基于 Win32 API 的未知病毒检测[J]. 计算机应用, 2008, 28(11): 2829-2831.
- [9] 张波云. 计算机病毒智能检测技术研究[D]. 长沙: 国防科学技术大学, 2007.
- [10] 徐燕, 李锦涛, 王斌等. 文本分类中特征选择的约束研究[J]. 计算机研究与发展, 2008, 45(4): 596-602.
- [11] 胡峰, 王国胤. 属性序下的快速约简算法[J]. 计算机学报, 2007, 30(8): 1429-1435.
- [12] 徐章艳, 刘作鹏, 杨炳儒, 等. 一个复杂度为 $\max\{O(|C||U|), O(|C|^2|U/C|)\}$ 的快速属性约简算法[J]. 计算机学报, 2006, 29(3): 391-399.
- [13] 刘勇, 熊蓉, 褚健. Hash 快速属性约简算法[J]. 计算机学报, 2009, 32(8): 1493-1499.

(上接第962页)

- [2] ZHANG JUN, LI JIEGU, ZHANG LING. Video watermark technique in motion vector [C]// Proceedings of XIV Symposium on Computer Graphics and Image Processing. Washington, DC: IEEE, 2001: 179-182.
- [3] XU CHANGYONG, PING XIJIAN, ZHANG TAO. Steganography in compressed video stream [C]// ICICIC'06. Washington, DC: IEEE, 2006: 269-272.
- [4] FANG DING-YU, CHANG LONG-WEN. Data hiding for digital video with phase of motion vector [C]// Proceedings of International Symposium on Circuits and Systems. Washington, DC: IEEE,

2006: 1422-1425.

- [5] 田丽华. 编码理论[M]. 2版. 西安: 西安电子科技大学出版社, 2007: 180-209.
- [6] 王新梅, 肖国镇. 纠错码——原理与方法[M]. 修订版. 西安: 西安电子科技大学出版社, 2006: 443-503.
- [7] 朱仲杰, 王玉儿, 蒋刚毅, 等. 基于自适应策略的稳健视频水印算法[J]. 计算机工程与应用, 2006, 36: 35-37.
- [8] HE XUANSEN, LUO ZHUN. A novel steganographic algorithm based on the motion vector phase [C]// International Conference on Computer Science and Software Engineering. Washington, DC: IEEE, 2008, 359: 822-825.