基于支持向量机的病毒程序检测方法

彭 宏[1],王 军[2]
(1. 西华大学计算机与数理学院,四川成都 610039;2. 西华大学电气信息学院,四川成都 610039)

摘 要: 支持向量机是一种对于小样本具有良好学习性能的机器学习方法. 本文将支持向量机方法用于病毒程序的检测中,可以改善其它方法在先验知识较少情况下的推广能力的问题. 仿真实验结果看出,该方法在训练样本数相对较少的情况下,仍然具有较高的检测率和正确率,同时也具有较低的虚警率.

关键词: 病毒程序; 恶意程序; 网络安全; 支持向量机; 统计学习
中图分类号: TP181 **文献标识码**: A **文章编号**: 0372-2112 (2005) 02-0276-03

Research of Malicious Executables Detection Method Based on Support Vector Machine

PENG Hong[1], WANG Jun[2]
(1. *School of Computer & Mathematical-Physical Science, Xihua University, Chengdu, Sichuan* 610039, *China*;
2. *School of Electric Information, Xihua University, Chengdu, Sichuan* 610039, *China*)

Abstract: Support vector machine is a machine study method with good performance when the sample size is small. The method of support vector machine is used to malicious executable in the paper that improves the generalizing ability with given less prior knowledge. Then simulation results express this method has better detection rate, overall accuracy and false positive rate reduced with less training sample size.

Key words: malicious executables; network security; support vector machine; statistical learning

1 引言

随着计算机网络的发展,尤其是互联网的迅速发展与普及,使得基于网络的计算机系统在我们的生活、工作、学习等诸多方面,正起着越来越大的作用. 但随之而来的各种计算机犯罪,尤其是基于网络的计算机病毒肆意传播,使得其所造成的危害也越来越大. 病毒恶意程序是一个执行危及系统安全、破坏系统或未经用户许可获取其敏感信息等功能的程序.

早期的恶意程序检测方法主要应用签字(特征)来帮助检测[1],这些签字(特征)包括很多不同的属性(如文件名、内容字符串或字节等),并且多从排除恶意程序所产生的漏洞的角度来获得系统的安全性. 遗憾的是,一个新的恶意程序可能不包括已知的特征. 目前每天大约产生 8 到 10 个病毒,其大部分是无法检测出来的,除非获得了它们的签字(特征). 恶意程序检测的一个关键问题是要使设计出的检测系统能够检测出没有见过的恶意程序. 为此提出了多种机器学习方法,用于恶意程序的检测. 如 RIPPER 方法[2]是一种归纳学习法,其产生基于特征属性的布尔规则用于检测今后的恶意代码. 多贝叶斯分类器[3]通过一组简单贝叶斯分类器的投票来确定一个实例的最终分类结果. 在文献[4]中讨论了神经网络的方法. 以上方法都需要大量或是完备的程序组成的数据集,并将这些程序分成恶意和普通两种类型,才能达到比较高的检测性能,并且训练时间较长.

支持向量机(SVM)是一种建立在统计学习理论之上的机器学习方法,其最大的特点是根据 Vapnik[5]结构风险最小化原则,尽量提高学习机的泛化能力,即由有限的训练样本得到小的误差可保证对独立的测试集的误差也很小. 另外,由于支持向量机算法是一个凸优化问题,所以局部最优解一定是全局最优解. 这是其它机器学习算法所不及的.

我们将支持向量机应用到恶意程序的检测中,在先验知识不足的情况下,可确保支持向量机分类器仍有较好的分类正确率,从而使得整个恶意程序检测程序具有较好的检测性能. 由此,我们提出了一种基于支持向量机的恶意程序检测模型.

2 特征选择

为了使用基于支持向量机的模型进行检测,首先需要从公开的资源中收集大量程序组成一个集合,并将这些程序分成恶意和普通的两种类型. 接下来需要自动从数据集的每个程序中提取二进制代码的概要描述,再从中抽取出分类学习所需要的特征. 利用这些特征训练分类器. 我们采用文献[3]的方法提取数据集的每个程序的不同特征,这些特征代表了各程序所包含的不同信息.

首先检查调用 LibBFD 的 PE(可移植)格式的程序,然后再利用更一般的方法从所有类型程序中抽取出相应的特征.

这里抽取的特征应能够描述程序代码的行为. 比如为了从 Windows 程序中提取有效的信息,可使用 GNU's Bin-Ttils 工具包. 该工具包能够帮助分析 Windows 下的 PE 格式程序,从

收稿日期:2004-02-09;修回日期:2004-06-26
基金项目:四川省教委自然重点科研项目(No. 0129844)

中可以抽取出对象格式信息:文件大小、所调用的动态连接库名称、动态连接库中调用函数名称和重定向表等,以形成一个特征向量.为了表征程序代码的行为,使用 LibBFD 从程序的头部信息中获得程序代码所调用的动态连接库中所有函数的信息.例如: *advapi*32. *AdjustTokenPrivileges ( )* ∧ *aviapi*32. *GetFileSecurityA ( )* ∧…∧*wsock*32. *recv ( )* ∧*wsock*32. *send ( )*.以上特征向量表示至少由四个资源组成,两个从 *advapi*32. *dll* 中调用函数 *AdjustTokenPrivileges ( )* 与从 *aviapi*32 中调用函数 *GetFileSecurityA ( )*;两个从 *wsock*32. *dll* 中调用函数 *recv ( )* 与 *send ( )*.

除此之外,还利用 GNU 字符串程序从数据集全部程序中提取有关特征.表 1 是利用 GNU 字符串程序抽取的一些频繁出现的字符串.在恶意程序中包含的这些特征字符串把它们与普通程序区分开.同样地,在普通程序中也存在与恶意程序相区分的字符串.代码中存在的每一个字符串都被作为一个特征.

当然这些字符串不是一种鲁棒的特征.因为它们能够被简单的修改,因此可通过分析另一个特征:字节顺序.字节信息特征包含了最大的信息量,因为它表现了一个程序的机器码,而不紧紧是 LibBFD 提供的资源信息.这可以借助诸如 hexdump 的工具来完成.这样每个字节顺序被作为有关特征.表 2 是利用 hexdump 的工具提取的字节码示例.

表 1　利用 GNU 抽取的一些字符串示例

Kernel	microsoft	windows
getversion	advapi	writefile
getprocaddress	messageboxa	regclosekey

表 2　利用 hexdump 抽取的一些字节码示例

6572	2073	694d	7263	736f	666f
646e	776f	2e73	0a0d	0024	0000
454e	3c05	026c	0009	0000	0000

3　支持向量机

支持向量机起源于统计学习理论,它研究如何实现模式分类问题[5].支持向量机使用结构风险最小化(简称 SRM 准则)原理构造决策超平面,使每一类数据之间的分类间隔最大.

支持向量机的思想就是在样本数目适宜的前提下,选取比较好的 VC 维 h,使经验风险 $Remp$ 和置信度达到一个折衷,最终使实际风险 R 变小.

对于两分类问题,存在线性可分和线性不可分两种支持向量机.但是在实际中,为了将两类模式尽可能分类开来,一般要构造非线性可分的支持向量机.Cover 定理指出:一个复杂的模式分类问题,在高维空间比低维空间更容易线性可分.支持向量机就是通过核函数把训练样本中的低维数据映射到高维特征空间,然后在高维特征空间构造一个最佳分类平面.

支持向量机中研究最多的核函数有 3 类:多项式核函数、径向基函数(RBF)和两层神经网络核函数.形式如下:

多项式核函数: $K(x,y) = (x \cdot y/256 + 1)^d$

RBF 核函数: $K(x,y) = \exp\left\{-\frac{|x-y|^2}{256\sigma^2}\right\}$

两层神经网络核函数: $K(x,y) = \tanh(ax \cdot y/256 - b)$

4　基于支持向量机的恶意程序检测模型

基于支持向量机的恶意程序检测模型主要由程序数据预处理器、支持向量机分类器和决策系统三部分组成,如图 1 所示.

为了进行基于支持向量机的恶意程序的检测,首先使用诸如 MacAfee 病毒扫描器将每个程序标记为恶意或普通(类别).接下来需要自动从数据集的每个程序中的二进制代码的概要描述中抽取所需的特征,以及利用 GNU 字符串程序抽取的一些频繁出现的字符串作为分类学习的特征.利用不同的特征来训练分类器.当然这些字符串不是一种鲁棒的特征.因为它们能够被简单的修改,因此可通过分析另一个特征:字节顺序.字节信息特征包含了最大的信息量,因为它表现了一个程序的机器码,而不紧紧是 *LibBFD* 提供的资源信息.这可以借助诸如 hexdump 的工具来完成.这样每个字节顺序被作为有关特征.

在模型中程序数据预处理器用来对收集的程序代码数据进行变换或处理.由于支持向量机的分类器只能对维数相同的数字向量进行分类,就必须把数据中不是数字类型的转换为数字向量.支持向量机对这些数字向量进行分类,产生判决结果.

图 1　基于支持向量机的恶意程序检测模型

模型中设立决策系统主要是为了提高模型的准确率.由于我们获得的训练样本的不完备,会导致恶意程序被错误标记为普通程序而普通程序错误标记为恶意程序.通过设立一些判别准则如百分比进行判别.

5　实验仿真

在我们的仿真实验中选用了文献[3]中的样本数据集进行仿真实验.在数据集中包含 4266 个程序,其中 3265 个为恶意程序,1001 个为普通程序,并且每个程序均通过病毒扫描器标记为恶意或普通(类别).

首先使用 GNU's Bin-Ttils 工具包对样本集的每个程序提取描述程序代码行为的有关特征信息,诸如:文件大小、所调用的动态连接库名称、动态连接库中调用函数名称和重定向表等.在恶意程序中包含的这些特征字符串把它们与普通程序区分开.同样地,在普通程序中也存在与恶意程序相区分的字符串.这些特征信息作为进行分析的特征向量构成训练数据样本集.

系统的工作过程分为两个阶段:训练阶段和检测阶段.在训练阶段,包含机器学习及测试两个步骤.首先根据已知的包含恶意程序与普通程序的样本集的训练数据来训练支持向量机,得到支持向量机相关参数,再根据测试结果对所选的核进行调整得到最优的学习模型.在检测阶段,首先将未知状态的程序特征数据处理成数字特征向量形式,然后通过支持向量机分类器对这些数字向量进行分类,并将结果提交决策系统作出最后的判别.

在试验中比较了 3 种常用的核函数的 SVM 算法的检测

结果(对应惩罚因子 C = 110).

表 3 多项式核函数对应的实验结果

d	1	2	3	4	5	6
平均 SV 个数	236	298	342	342	298	298
平均检测率(%)	97.68	95.83	95.45	94.79	94.63	94.12

表 4 RBF 核函数对应的实验结果

σ^2	0.1	0.2	0.4	0.8
平均 SV 个数	236	146	184	184
平均检测率(%)	96.83	97.63	98.13	96.28
σ^2	1.0	1.2	2.0	4.0
平均 SV 个数	184	184	198	236
平均检测率(%)	96.15	96.42	96.23	95.93

表 5 两层神经网络核函数对应的实验结果

$a = 1$	$b = 0.8$	$b = 0.9$	$b = 1.0$	$b = 1.1$	$b = 1.2$
平均 SV 个数	236	236	236	298	298
平均检测率(%)	96.43	95.83	9524	94.81	94.46
$a = 2$	$b = 0.8$	$b = 0.9$	$b = 1.0$	$b = 1.1$	$b = 1.2$
平均 SV 个数	184	184	184	198	236
平均检测率(%)	97.59	97.23	96.87	96.31	95.68

试验结果表明:对于多项式核函数,参数 $d = 1$ 时,最大检测率为 97.68 %,且随着参数 d 的增大,检测率有所下降.对于径向基函数,在参数 $\sigma^2 = 0.4$ 时,最大检测率为 98.13 %,随着 σ^2 的增大,SV 个数增加.对于两层神经网络核函数,在参数 $a = 2$, $b = 0.8$ 时,最大检测率为 98.13 %,随着参数 b(a 固定)的增大,SV 个数增加.

对于惩罚因子 C 的选取,直接影响分类器的推广能力.我们在实验中选取多项式核函数(参数 $d = 1$),径向基函数($\sigma^2 = 0.4$),两层神经网络核函数($a = 2$, $b = 0.8$)作为核函数,分别取不同的惩罚因子 $C = 0.1$, $C = 1$, $C = 10$, $C = 110$, $C = 500$, $C = 1000$,进行试验.试验表明,对于不同的核函数,C 的最优取值不同.随着 C 的增大,SV 个数将减少,且 SV 个数与检测率趋于稳定值.其中当选取径向基函数在惩罚因子 $C = 110$ 时,检测率取最大值 98.13 %.

为了将本文提出的方法与文献[3]中方法进行比较,这里使用文献[3]的几个量化指标:TP、TN、FP、FN、检测率、虚警率和正确率.在实验中采用 5 次交叉验证法来得到我们的实验结果.交叉验证方法是一种评估对未知数据预测准确率的标准方法.5 次交叉验证法是将整个(程序)样本数据集划分为五等份,利用其中四份进行模型训练,利用最后一份进行测试.如此循环五次,每次均留下一个未使用个的数据集进行测试.对这五次求平均,就可以获得一个(对未知数据)较为可靠的准确率.文献[3]给出的结果采用如上 5 次交叉验证法.为了与文献[3]相比较,说明支持向量机方法处理小样本的能力,在仿真实验中我们采用将整个(程序)样本数据集划分为五等份,利用其中一份进行模型训练,利用剩下的四份进行测试.

表 6 给出本文的仿真结果(采用径向基函数:$C = 110$ 与 $\sigma^2 = 0.4$)与文献[3]中研究结果进行比较.表 6 中的前三个方法的结果来自于文献[3].从实验结果可以看出,基于支持向量机分类器的方法远高于基于签字和 RIPPER 方法.和多贝叶斯方法比较,在训练样本数相对较少的情况下,仍然具有较高的分检测率和算法的正确率,同时也具有较低的虚警率.

表 6 几种检测方法的结果比较

	TP	TN	FP	FN	检测率	虚警率	正确率
基于签字方法	1102	1000	0	2163	33.5 %	0 %	49.28 %
RIPPER 方法							
—DLL	22	187	19	16	57.89 %	9.22 %	83.62 %
—DLL 函数	27	190	16	11	71.05 %	7.77 %	89.36 %
—DLL 函数数量	20	195	11	18	52.63 %	5.34 %	89.07 %
多贝叶斯方法	3191	940	61	74	97.76 %	6.01 %	96.88 %
支持向量机方法	3188	938	63	77	97.64 %	6.29 %	96.71

6 结论

本文提出的将支持向量机用于病毒程序的检测方法,可以改善其它方法在先验知识较少情况下的推广能力的问题.仿真实验表明,该方法在训练样本数相对较少的情况下,仍然具有较好的效果.

参考文献:

[1] Steve R W, Morton S, et al. Anatomy of a Commercial-Grade Immune System[R]. IBM Research White Paper, 1999.

[2] William C. Learning trees and rules with setValued features[A]. American Association for Artificial Intelligence (AAAI)[C]. Menlo Park, CA: AAAI Press, 1996. (1). 709 - 716.

[3] Matthew GS, Eleazer E, et al. Data mining methods for detection of new malicious executables[A]. IEEE Symposium on Security and Privacy[C]. Oakland, CA, May 2001: 1207 - 1217.

[4] Gerald T, Jeffrey O, et al. Neural networks for computer virus recognition[J]. IEEE Expert, IEEE Computer Society, August, 1966, 11(4): 5 - 6.

[5] Vapnik V N. The Nature of Statistical Learning Theory[M]. New York: Spring-Verlag, 1995.

作者简介:

彭 宏 男,1966 年 7 月出生于四川犍为,副教授,主要从事信息安全,数据挖掘和智能信息处理等方面的研究,发表论文 20 余篇. E-mail: jiayu@mail.sc.cninfo.net.

王 军 女,1966 年 12 月出生于四川绵阳,教授,主要从事智能控制和智能信息处理等方面的研究,发表论文 20 余篇.