用于行为分析反木马的模糊分类算法研究

陈庆章　莫建华　顾雨捷

浙江工业大学信息学院，杭州 310014

摘　要: 目前木马检测方法大多为特征码检测技术体系.不能够检测特征码未知的新木马.行为分析反木马技术具备了反未知木马的能力.但对现有基于行为分析技术的反木马策略分析后发现,大多存在着误报,漏报率过高,应用效率较低,交互不合理等问题.本文通过对反木马算法理论体系研究和木马行为特征分析,建立了一套反木马算法设计标准,为算法的构建和实际应用提供有力的依据.在此基础上提出并设计了用于行为分析反木马的模糊分类算法,最后利用实验证明本文反木马算法的有效性.

关键词: 木马 行为分析 模糊分类 分类精度

1 引　　言

目前木马的数量上升之快，木马的制作手段之成熟，木马的危害之大，使得占主导地位的特征码检测技术已经无法应对来自木马日益增长的威胁。现有的对抗特征码技术的木马免杀方法非常之多。利用这些免杀办法，经过修改过的木马就成为了一个具有新的特征串序列的未知木马了。可以看出，上述免查杀的方法对黑客技术要求较低，木马的生产速度较快，免杀的效率和效率上均达到了较好的水平。使用上述方法每产生一个木马，反木马厂商就要为该木马产生一个特征串序列，加入到特征库，提供用户下供更新。这样一个防御性的过程在速度远远比不上产生一个具有新的特征码的木马来得迅速便捷。因此，防御滞后于攻击的现象越来越明显，以至于单一的特征码检测技术已经无法很好的保护我们的信息安全。这样我们有必要引入一种防御未知木马的新检测方法。

行为分析反木马技术就是根据程序的动态行为特征（如在注册表设置自启动项等）判断其是否可疑。行为分析这种技术良好地解决了网络安全业界防御滞后于攻击的现象，实现了主动防御的思想。与特征码检测技术相比，木马的“行为特征库”更容易被归纳总结出来，同时，由于不会像“特征码库”那样占用大量空间并且频繁增长，所以更容易维护，开销更小，理论上讲用在反木马中检测木马的效率也更高。

2 数 据 抽 象

首先需要将木马与合法程序的样本抽象模糊分类系统的输入形式。若现有 p 个已知木马和 q 个合法程序（p>0，q>0 并且 p 和 q 均为整数），则样本空间记为 n = (p+q)。确定 m 个行为特征变量。分类数为 2 类。则将 n 个样本空间记为：

$$X=\{x_1,x_2,\cdots,x_n\}$$

每个样本用 m 个指标特征向量表示：

$$x_j=\left(x_{1j},x_{2j},\cdots,x_{mj}\right)^T$$

则样本集可用 m*n 阶指标特征值矩阵表示：

$$x_{m\times n}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}=(x_{ij})$$

x_{ij} 为样本j指标i的特征值，i = 1,2,…,m;,j = 1,2,…,n；

由于 m 个指标特征值物理量的量纲不同，在进行识别时要先消除指标特征值量纲的影响，使指标特征值规格化，即将指标特征值矩阵变换为指标特征值的相对隶属度矩阵：

$$R=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}=(r_{ij})$$

其中，r_{ij} 为指标特征值规格化数或相对隶属度，$0\le r_{ij}\le 1$。

设将 n 个样本依据样本的 m 个指标特征按 C 个级别(或类别)加以识别，其模糊识别矩阵为：

$$U=\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{c1} & u_{c2} & \cdots & u_{cn} \end{bmatrix}=(u_{hj})$$

其中，u_{hj} 为样本 j 从属于级别 h 的相对隶属度，h = 1,2…C,满足条件：

$$\begin{cases} \sum_{h=1}^{c} u_{hj}=1, & \forall j \quad 0\le u_{hj}\le 1 \\ \sum_{j=1}^{n} u_{hj}>0, & \forall h \end{cases}$$

设每个级别 h 有 m 个指标特征值（称为标准指标特征值），则 C 个级别的指标特征可用 m*C 阶标准指标特征值矩阵表示：

$$Y=\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & y_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mc} \end{bmatrix}=(y_{ih})$$

式中 y_{ih} 为级别 h 指标 i 的标准指标特征值。

计算各特征属性的隶属度，我们统计各特征属性（即 A_i）在每一木马程序 v_j 中出现的频率 A_{ij}^{v} 及在每一正常程序 N_j 中出现的频率 A_{ij}^{N}。然后，我们可以得到对应的特征属性的均值 $E(A_i^v)=\frac{1}{s}\sum_{i=1}^{s}A_{ij}^{v}$，s 为木马样本数量，同样，正常程序的行为属性的均值为 $E(A_i^N)=\frac{1}{n}\sum_{i=1}^{n}A_{ij}^{N}$。总频率为 $E(A_i)=\frac{E(A_i^v)+E(A_i^n)}{2}$。

利用 F 分布中的正态偏大型模糊分布来构造木马程序集 V 的隶属函数：

$$\mu_V(E(A_i^V))=\begin{cases} 0, & E(A_i^V)<0 \\ 1-e^{-(E(A_i^V))^2/\sigma^2}, & E(A_i^V)\ge 0 \end{cases} \tag{1}$$

式中：$\sigma = \max\{E(A_1^v), E(A_2^v), ..., E(A_7^v)\}/7$。

同样，构造合法程序集 N 的隶属函数为：

$$\mu_N(E(A_i^N)) = \begin{cases} 0, & E(A_i^N) < 0 \\ 1 - e^{-(E(A_i^N))^2/\sigma^2}, & E(A_i^N) \geq 0 \end{cases} \tag{2}$$

对于待检测程序文件，其隶属函数为：

$$\mu_M(A_i) = \begin{cases} 0, & A_i < 0 \\ 1 - e^{-(A_i)^2/\sigma^2}, & A_i \geq 0 \end{cases} \tag{3}$$

从而，我们可以计算出样本的隶属度，从而样本表示成模糊集形式，分种形式分别为：

木马模糊表示：$\tilde{V} = \{\mu_1^V / A_1, \mu_2^V / A_2, \cdots, \mu_t^V / A_t\}$

合法程序模糊表示：$\tilde{N} = \{\mu_1^N / A_1, \mu_2^N / A_2, \cdots, \mu_t^N / A_t\}$

待测程序模糊表示：$\tilde{M} = \{\mu_1 / A_1, \mu_2 / A_2, \cdots, \mu_t / A_t\}$

至此为止，样本数据已经抽象成了算法可以直接输入的数据。这样，通过以上处理数据形成了算法可以输入的样本空间。

3 分类器模型设计

基于自适应调节机制的模糊分类器总体上分为三个部分：第一部分是一个多层模糊分类系统用于建立最初的 IF-THEN 规则，每条规则带有一个置信度值。紧接着是分类单元块，产生可解释的 IF-THEN 规则。第三则是自适应调节机制模块，用于调节规则的强度。图 1 是这个模型的总体设计图。

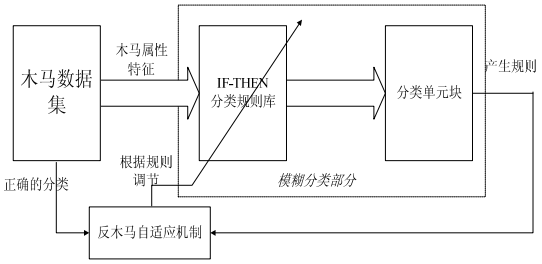

图 1 新的反木马算法的模型

其中，数据集部分是训练数据的样本空间，数据在这里已经做了抽象处理，将数据表示成为了可输入的模糊形式。下一步数据流经虚框内，也就是分层模糊分器部分，该部分分为两个部分，分别是 IF-THEN 规则库，用于产生规则以及分类单元块，用于产生分类结果。对分类后的结果进行评判由自适应机制模块进行，用于对分类正确与错误的规则进行进一步的置信值调节，使其在下一次分类的时候更加精确。下面着重对多层分类器和自适应机制进行详细描述。

4 多层模糊分类模块

构建一个多层分类器是数据处理的第一步，在这里，我们为了引入自适应机制。

定义一个有输入-输出的数据集，存在 n 个特征变量。用函数表示为 S：$R^n \to R$。用向量表示则为：$x_p = [x_{p1}, x_{p2}, \cdots, x_{pn}], p = 1, 2, ..., m$。比如在文中，我们收集了 400 个程序，每个程序提取出了 7 个特征变

量，最终分类木马程序与合法程序两个输出类。在分类单元部分的模糊 IF-THEN 规则 R_a^k 描述如下：

IF x_{p1} is $A_{j1}(x_{p1})$ and x_{p2} is $A_{j2}(x_{p2})$ … x_{pn} is $A_{jn}(x_{pn})$ THEN x_p 属于类别 M 的置信值为 CV_a^k。其中：$A_{jn}(x_{pn})$ 表示前提条件的隶属度值空间；a 表示规则数；K 表示子区间号；i 表示特征属性的索引下标；j 表示子区间号的索引下标。

第一步，子区间{ A_1^k，…， A_k^k }的隶属度划分采用三角隶属度函数，$A_j^K(x_{pi}) = \max\{1 - \frac{|x_{pi} - a_j^K|}{b}, 0\}$，其中：b=2/(K - 1)， a_j^k = -1+(j - 1)b ,j = 1,2,…,k。

在这里，我们设 ε_{ij}^k 为属性 x_{pi} 到区间 A_j 的中心距离。如果隶属值为 1，则属性特征的距离值为 0。相反，如果属性特征完全不属于 A_j，那么设 ε_{ij}^k 为一个无限大的数，这里使用 10000。

则有距离公式： $$\varepsilon_{ij}^K = (1 - \mu_{A_j^K}(x_{pi}))\, jump(\mu_{A_j^K}(x_{pi}))$$

$$jump(\delta) = \begin{cases} 1, & \delta > 0 \\ 10000, & \delta = 0 \end{cases}$$

然后我们需要计算每条规则的类贴近度：

$$\psi_{ClassT}(R_a^K) = \sum_{x_p \in ClassT} \frac{1}{\sqrt{\sum_{i=1}^{n} (\varepsilon_{ija}^K)^2}}$$

如果 $\psi_{ClassT}(R_a^K)$（例如 T = 1）大于其它的类别，则表示该规则对应的样本属于类别 1 的情况多于同等情况下其它的类别。如果 $\sum_{i=1}^{n} (\varepsilon_{ija}^K)^2$ 等于 0，对应的值就是无限大，则表示 R_a^K 完全地属于对应的类别。也就是说，至少有一个样本在这个规则之中。

第二步，为每条规则 R_a^K 找到最大贴近类

$$T = \arg \underset{M}{Max}\{\psi_{Class1}, \psi_{Class2}, \ldots, \psi_{ClassM}\}$$

该公式表示规则属于 ψ_{ClassT} 值最大的类。当存在两个或两个以上的最大值并列时，或者 ψ_{ClassT} 值为 0，则表示这条规则的 IF-THEN 的结论部无法唯一地确定，我们视该规则将视为一条冗余规则，T = 0。也就是说，这条规则不能用于分类判别。

第三步，如果第二步中的公司的结果 T 不为 0，则可以为该公式计算置信值：

$$CV_a^K = \frac{Max\{\psi_{Class1}, \psi_{Class2}, \ldots, \psi_{ClassM}\}}{\sum_{T=1}^{M} \psi_{ClassT}}$$

对于 $T \neq X$，如果分母 ψ_{ClassT} 为 0，ψ_{ClassX} 大于 0，那么 CV_a^K = 1，这表示所有包含这条规则的样本都必然属于同一个类。如果公式 3.6 值为 0，那么 CV_a^K = 0。

最后一步我们还需要来计算类的置信水平（confidence level）σ_{ClassT}，用公式：

$\sigma_{ClassT} = \max\{\alpha_P \cdot CV_a^K \,|\, R_a^K \in S, R_a 部分结果 \in ClassT\}$ 来计算。其中，$\alpha_P = \frac{1}{\left\| \sum_{i=1}^{n} (\varepsilon_{ij}^K)^2 \right\|}$，S 是模型中建立的规则库。

一个样本通常是同时要包含于多条规则的，一条规则又常常可以导出多个分类并存在多个置信值。因

此，当有多条规则可供选择时，我们用 $\alpha_P \cdot CV_a^K$ 这个乘积值最大的规则来选择规则。

5 自适应机制模块

在上述计算过程的基础上我们引入了自适应机制，通过设置参数并且不断得调整置信值来获得更高的分类精度。

在自适应机制中，规则会得到一个动态的变化值。如果一个输入分类正确，那么对应的参数增加表示奖励。由于该规则对样本的正确分类，规则的置信值同时得到提高。新的置信值用以下公式计算：

$$CV_t^{New} = CV_t^{old} + w_1 \cdot (1 - CV_t^{old})；\ w_1 = \eta_1 \cdot \frac{\sigma_{ClassT} - \underset{M \neq T}{Max}\{\sigma_{Class1}, \sigma_{Class2}, \ldots, \sigma_{ClassM}\}}{\sum_{m=1}^{M} \sigma_{ClassM}}$$

同样，如果样本的输入造成了错误的分类，我们一样有公式

$$CV_t^{New} = CV_t^{old} - w_2 CV_t^{old}；\ w_2 = \eta_2 \cdot \frac{\sigma_{ClassT}}{\sum_{m=1}^{M} \sigma_{ClassM}}$$

来计算分类错误之后规则置信值的变化。对应的置信值表示一种“惩罚”机制，由于置信值不能为负，因此最低为 0。同样，最高不能超过 1。

6 实验与性能分析

我们建立起实验环境以测试算法效果。其中，模糊子系统的搭建上我们采用了 matlab 里 toolbox 中的模糊逻辑（fuzzy logic）来辅助建立。自适应模块的计算，我们利用 matlab 编程进行数值计算来对置信值进行增或减。最后完成了总体模糊上的数据计算。实验样本上，我们收集了共 400 个程序，其中，200 个合法程序与 200 个木马样本。

实验中首次引入了交叉验证解决了木马测试环节长期存在的小数据量和数据权威性等问题，确保测试的效果。实验结果表明，本文的构建在模糊模型上的反木马算法在训练阶段通过自适应机制在有效迭代下收敛到了 100%的精度，并且在测试阶段达到了较高的精度，该精度在同等实验条件下与文献[63]中的贝叶斯分类算法的结果比较发现，在算法设计更为全面的基础上，分类精度上也有明显提高。

为了进行实验，共收集了 200 个合法程序和 200 个木马样本的技术描述，其中 200 个合法程序由网上下载得到。限于实验条件没有办法对木马样本进行测试，因此我们从著名信息安全厂商 Symantec 官方网站的木马库中得到木马的技术细节描述作为我们行为特征提取的依据。具体的查询网址是：http://www.symantec.com/business/security_response/index.jsp。

我们提取了七个常见的行为特征：(1)在注册表设置自启动项、(2)拷贝文件到系统目录下(或在该目录下创建文件)、(3)呈现可视界面、(4)注射远程线程到其它进程、(5)在注册表修改某些文件关联、(6)网络通信时绑定某个网络端口、(7)杀死其它进程

我们将收集得到的实验样本按照上述 7 个行为特征查看木马及合法程序是否呈现了上述行为特征。其中，鉴于相关法律法规限制，木马的行为特征是 Symantec 官方网站提供的样本描述中自行分析得到。合法程序不仅从软件描述中分析得到，并且利用了一系列 API 调用跟踪程序来辅助分析，例如 Windows 平台下的软件自动调试监视工具 Auto Debug for Windows。

我们在实验采用训练数据本身作为测试数据的方法（training-data-itself-as testing），也就是取出一部分训练数据来作为测试数据，与公开的模式识别的样本库(ftp:ftp.ics.uci.edu/pub/machine-learning-databases)不同，木马以及合法程序的样本库至今尚未形成一个权威的数据集以用于测试算法的效果。利用交叉验证充分利用了所有已有的样本，很大程度上解决了小样本空间问题以及样本的随意性带来的实验偏差，保证在

将来的未知数据集上的最佳预测精度。

我们采用 4 次交叉验证，将数据分成四组，其中四组分别含有木马样本 50 个，合法程序 50 个。分别编号为组一，组二，组三，组四。组一的交叉验证的训练阶段使用组二，组三，组四进行训练，保留组一用于测试阶段。同样，组二，组三，组四的情况类似。

训练阶段中，11 轮迭代的数据后，分类的精度均达到了 100%。测试阶段中，每组分别使用未使用的组来进行测试，我们得到组一的分类精度为 88.32%，组二的分类精度为 94.65%，组三的分类精度为 92.93%，组四的分类精度为 81.34%。总体上看，最佳分类精度为 94.65%，最差分类精度为 81.34%，平均分类精度为 89.31%。由于贝叶斯分类算法在反未知恶意代码上应用是最早被哥伦比亚大学的专家提出，沿用比较广泛，具有典型性。贝叶斯反木马算法的分类结果与我们的实验数据比较结果如表 1。

表 1 与贝叶斯分类算法的结果比较

精度	反木马模糊分类算法	反木马贝叶斯分类算法[63]
平均精度	90.81%	88.5%
最佳精度	94.65%	88.5%
最差精度	87.34%	88.5%

我们发现，我们的算法在平均精度上高了约 2 个百分点，达到了百分之九十以上，最佳精度高了贝叶斯算法约 6 个百分点，最差精度比贝叶斯略低。另外在各组的交叉中我们发现，四组精度值的抖动比较小，表明引入的自适应机制收敛效果比较理想。

参 考 文 献

[1] Fred Cohen. Reply to Comment on A Framework for Modelling Trojans and Computer Virus Infection by E.Makinen[J]. The Computer Jouranl, vol.44. No 4. 2001. pp. 326-327.ISSN 1460-2067.

[2] Kim J, Bentley P. AnArificail Immune Model for Network Intrusion Detection[J]. In:Proc.of 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen, Germany, 1999.

[3] 商海波. 一种基于行为分析的反木马策略[D]. 学位论文, 2005.

Research on Classification Algorithms of Trojan Horse Detection Based on Behavior

Chen Qing-zhang Muo Jian-hua Gu Yu-jie

Department of Information engineering, Zhejiang University of technology, HangZhou, 310014

Abstract: Current anti-Trojan is almost signature-based strategies, which cannot detect new one. Behavior analysis, with the ability to detect Trojans with unknown signatures, is a technique of initiative defense. However Current behavior analysis based anti-Trojan strategies have the following problems: high false based anti-Trojan strategies have the following problem: high false or failure alarm rate, poor efficiency, and poor user-friendly interface design, etc. The paper works on the design of an anti-Trojan oriented algorithm based on behavior analysis. And we construct standard of anti-Trojan algorithm system and point the up-limit of the precision. Finally, we organize the experiment to get the result.

Keywords: Trojan horse, behavior analysis; fuzzy classification; classification rate