第 34 卷 第 3 期 计 算 机 工 程 2008 年 2 月
Vol.34 No.3 Computer Engineering February 2008
·安全技术· 文章编号：1000—3428(2008)03—0154—03 文献标识码：A 中图分类号：TP309

基于 RST 修正核单类 SVM 的程序行为控制系统

骆玉霞[1,2]，刘金刚[1,3]
（1. 中国科学院计算技术研究所，北京 100080；2. 中国科学院研究生院，北京 100039；
3. 首都师范大学计算机科学联合研究院，北京 100037）

摘 要：程序行为控制系统对程序行为进行建模、检测和响应。单类支持向量机(SVM)在有限样本的情况下用于异常检测，具有较好的分类精度和泛化能力。针对以前利用单类支持向量机进行异常检测的研究中没有考虑属性权重的问题，该文提出利用粗糙集理论(RST)，引入反映属性重要性程度的权重值。给出通过找出决策系统中所有约简的集合确定属性权重的方法，并利用属性权重修正单类 SVM 的核函数。实验表明基于 RST 修正核的单类 SVM 具有更好的检测能力。

关键词：粗糙集理论；单类支持向量机；程序行为控制；异常检测

Program Behavior Control System Based on One-class SVM with Adjusted Kernel Using RST

LUO Yu-xia[1,2], LIU Jin-gang[1,3]
(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080; 2. Graduate School of Chinese Academy of Sciences, Beijing 100039; 3. Joint Faculty of Computer Science, Capital Normal University, Beijing 100037)

【Abstract】Program behavior control system includes program behavior model, detection and response. One-class Support Vector Machine (SVM) has good classification accuracy and generalization in the case of limited samples. In former research of anomaly detection using one-class support vector machine, the attribute weights are not considered. This paper presents a method to introduce weights that reflect attribute importance using Rough Set Theory (RST). The kernel function of one-class support vector machine is adjusted by calculating attribute weights though finding all reducts in the decision system. Experimental results show that the one-class support vector machine with adjusted kernel function using rough set theory has more effective detection capability.

【Key words】rough set theory; one-class support vector machine; program behavior control; anomaly detection

1 概述

程序行为控制是一种主动安全机制，通过对程序行为建模，监控其行为是否符合行为模型，如发现异常则采取相应的响应措施。程序行为控制结合了访问控制和入侵检测的特征。

访问控制可分为强制访问控制(mandatory access control)和自主访问控制(discretionary access control)。访问控制模型有 Bell-LaPadula 模型、Lattice 模型、Biba 模型、Clark Wilson 模型和 Chinese Wall 模型等。在访问控制系统中，通过定义主体和客体的关系来进行控制。程序行为控制监控进程的系统调用，其监控粒度比访问控制细。

入侵检测可分为异常检测和误用检测。Forrest 等人最先提出对特权进程的系统调用进行实时监测和分析，Hofmeyr 等人提出了对短序列的修正方法，另外还有数据挖掘的方法、马尔可夫模型和哈明距离等方法。程序行为控制与入侵检测的共同之处在于都对程序的执行进行检测，以发现入侵和异常程序行为。与入侵检测的不同，程序行为控制不但对程序的未知行为进行检测，而且具有访问控制的特征，可以保证程序能够按照预期设计的方式运行，有效地提高系统的安全性。

在实际应用中，通常只能获得正常的训练样本，异常的训练样本(攻击数据)很难得到，而且正常训练样本往往也是不完备的。在正常训练样本不完备的情况下，以往的研究方法会产生大量的误报。对于有限样本的情况，支持向量机(Support Vector Machine, SVM)具有较好的分类精度和泛化能力。单类支持向量机提供仅需利用正常数据进行训练的非监督分类算法，可以用于异常检测。

以前利用单类支持向量机进行异常检测的研究没有考虑属性的权重问题，其核函数中所有属性对决策起的作用相同。正确确定属性的权重，对于提高分类的正确性具有重要意义。粗糙集理论(Rough Set Theory, RST)提供了一套较完备系统的方法，用于从小样本数据中发现规律，通过分析样本数据集，能发现在数据集中的数据属性间的关系，计算出反映数据之间本质关系的所有属性的重要性关系[1-2]。

本文利用 RST 修正单类 SVM 的核函数，引入反映属性重要性程度的权重值，实验表明提高了单类 SVM 的检测能力。

2 单类 SVM 和 RST 原理

2.1 单类 SVM 原理

支持向量机是 Vapnik 等人提出的应用于监督分类的线

作者简介：骆玉霞(1975－)，女，博士研究生，主研方向：信息安全，数据挖掘；刘金刚，教授、博士

收稿日期：2007-02-10 **E-mail：**luo_yu_xia@163.com

性分类器的一种设计最佳准则，可以扩展到线性不可分和使用非线性函数的情况。两类线性可分的训练样本集之间存在一个隔离带，设 H 为平分面，处在隔离带的边缘位于与 H 平行的分界面上的样本点决定了隔离带，这些样本点称为支持向量。SVM 的主要目的是构造满足最大间隔准则的分界面。

针对非监督分类的问题，scholkopf 提出单类支持向量机算法。对于样本数据集 $\chi=\{x_i \mid i=1,\cdots,l; x_i=\{x_{i1},\cdots,x_{id}\}\}$，其中，$l$ 为样本数；d 为样本维数。通过特征映射 $\Phi:\chi\to F$，在高维空间中计算一个最小超球作为决策边界，使得大多数的样本点在以 a 为中心，半径为 R 的小球区域内，分类为+1，其余样本点的分类为-1，从而形成原问题为

$$\min\left(\frac{1}{2}R^2+\frac{1}{vl}\sum_{i=1}^{l}\xi_i\right) \tag{1}$$

其中，ξ_i 为缓冲量；$v\in(0,1]$。约束条件为 $(\Phi(x_i)-a)(\Phi(x_i)-a)^{\mathrm{T}}\le R^2+\xi_i$，$\xi_i\ge 0, i=1,\cdots,l$。

为求解原问题，引入 Lagrange 系数 a_i，核函数 $k(x_i,x_j)=(\Phi(x_i)\cdot\Phi(x_j))$，可将问题转化为对偶问题：

$$\min\left(\sum_{i=1}^{l}\sum_{j=1}^{l}a_ia_jk(x_i,x_j)-\sum_{i=1}^{l}a_ik(x_i,x_i)\right) \tag{2}$$

约束条件为 $0\le a_i\le\frac{1}{vl}$，$\sum_{i=1}^{l}a_i=1$，$i=1,\cdots,l$。求解该问题，使得决策函数：

$$f(x)=\mathrm{sgn}\left(R^2-\sum_{i=1}^{l}\sum_{j=1}^{l}a_ia_jk(x_i,x_j)+2\sum_{i=1}^{l}a_ik(x_i,x)-k(x,x)\right) \tag{3}$$

对于训练集中的大多数样本为正，同时又使得球的半径 R 很小。Lagrange 系数 a_i 不为 0 的 d 维样本 $x_i=\{x_{i1},\cdots,x_{id}\}$ 就是支持向量。

2.2 RST原理[3-4]

定义 1 信息系统是一个系统$I=(U, A)$，其中，$U=\{u_1, u_2,\cdots,u_{|U|}\}$是一个有限非空集，称为论域，$U$中的元素称为对象；$A=\{a_1,a_2\cdots,a_{|A|}\}$也是一个有限非空集，$A$中的元素称为属性，对于每个$a\in A$，有一个映射$a:U\to a(U)$，且$a(U)=\{a(u)|u\in U\}$称为属性的值域。

定义 2 如果信息系统 $I=(U, A)$中，$A=C\cup D, C\cap D=\phi$，则称信息系统为决策表，其中，C 中的属性称为条件属性；D 中的属性称为决策属性。

对于一个进程的正常系统调用序列，用长度为 d 的滑动窗口在系统调用序列上滑动，得到有 d 个属性的对象构成的决策系统，d 个属性中前 d-1 个属性作为条件属性，最后一个属性作为决策属性。

定义 3 信息系统 I=(U,A)中对于属性子集 $B\subseteq A$，式(4)定义的关系为不可分关系。

$$IND(B)=\{(x,y)\in U^2,\forall b\in B, b(x)=b(y)\} \tag{4}$$

定义 4 约简是指决策系统 $I=(U,A)$中，使得 $IND(B)=IND(A)$成立的最小属性子集 B, B 能够区分用整个属性集合 A 可区分的所有对象。约简具有不唯一性，可能有多个。

3 单类 SVM 的 RST 修正核

本研究中单类 SVM 的核函数采用高斯径向基函数作为核函数，其形式如式(5)。

$$k(x_i,x_j)=\exp\left(-\frac{|x_i-x_j|^2}{\sigma^2}\right)=\exp\left(-\frac{(x_i\cdot x_i)+(x_j\cdot x_j)-2\times(x_i\cdot x_j)}{\sigma^2}\right) \tag{5}$$

原核函数中，没有考虑样本属性对于决策的重要性问题，所有样本属性的权重相同，正确确定属性的权重，对于正确的决策具有重要意义。粗糙集理论提供了一套较完备系统的方法，用于从小样本数据中发现规律，通过分析样本数据集，能发现在数据集中的数据属性间的关系，计算出反映数据之间本质关系的所有的属性的重要性关系。本研究在计算样本的点积时，引入反映属性重要性程度的属性权重值，核函数中样本 x_i 和 x_j 的点积利用式(6)进行计算。

$$x_i\cdot x_j=\sum_{k=1}^{d}(x_{ik}\times x_{jk}\times w_k) \tag{6}$$

式(6)中属性权重的集合$W=\{w_1,\cdots,w_k,\cdots,w_d\}$利用粗糙集理论来确定。计算属性权重的方法是：(1)利用粗糙集理论找出决策系统中所有约简的集合$RED=\{R_1,R_2,\cdots,R_n\}$；(2)条件属性的权重等于包含该属性的约简个数；(3)决策属性的权重等于最大的条件属性权重加 1。确定属性权重集合的具体方法见算法 1。

算法 1 确定核函数中属性特征的权重

输入 决策系统 D

输出 属性权重的集合$W=\{w_1,\cdots,w_k,\cdots,w_d\}$

方法：

(1)计算决策系统D的所有约简集合RED＝{R_1, R_2,⋯,R_n }。
(2) for 每一个属性的权重w_i do begin
(3) 初始化属性的权重w_i为 0;
(4) end
(5) for 每一个约简R_i do begin
(6) for 约简R_i中的每一个属性j do begin
(7) w_j += 1;
(8) end
(9) end
(10) $w_d=\max(w_1,w_2,\cdots,w_{d-1})+1$;

4 程序行为控制系统

基于 RST 修正核单类 SVM 的程序行为控制系统的主要思想是，利用基于 RST 修正核单类 SVM 分类器来进行检测，通过(Loadable Kernel Module, LKM)拦截系统调用来实现程序行为控制，目的是有效地提高系统的安全防护能力，其系统模型如图 1 所示。程序行为控制系统分 3 个阶段进行：(1)训练阶段(training phase)；(2)检测阶段(detection phase)；(3)响应阶段(response phase)。如图 1 所示，程序行为控制系统的工作机制为：对正常系统调用序列组成的决策系统，采用粗糙集理论计算决策系统的约简，利用约简计算属性权重，从而得到单类支持向量机的 RST 修正核函数，通过对正常进程痕迹进行训练，建立基于 RST 修正核的单类 SVM 模型；对于未知的程序执行，利用可加载内核模块拦截系统调用，使用滑动窗口生成系统调用序列，然后使用基于 RST 修正核的单类 SVM 分类器进行分类，对于发现的异常进行记入日志、不执行系统调用返回继续执行和终止进程几种自动响应。

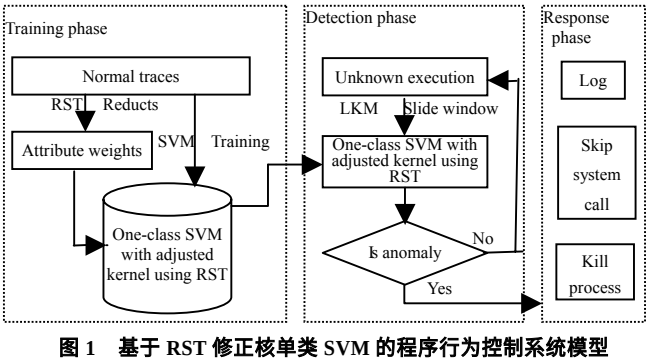

图 1 基于 RST 修正核单类 SVM 的程序行为控制系统模型

LKM 是 Linux 系统用于扩展内核功能的一种机制。LKM 允许具有超级用户权限的用户在运行着的系统上动态地增加或删除功能模块，为内核增加新的功能。LKM 的优点在于，其程序编写、编译、加载和卸载过程中，无须对内核进行重新编译就可以完成。由于 LKM 作为内核模块运行，因此使用 LKM 机制可以在内核空间有效地加强系统的安全防护能力。

在用户空间实现的入侵检测系统往往只能将入侵记入日志、进行事后分析，而不能阻止入侵的发生，与之相比，基于 RST 修正核单类 SVM 的程序行为控制系统，能够拦截系统调用，在系统调用发生之前判断是否为异常行为，能够阻止异常的发生，同时，该系统根据异常行为的危害程度，可以进行多种方式的自动响应。

5 实验与分析

本研究利用 MIT 人工智能实验室提供的 lpr 数据进行实验。数据集中有 2 766 个正常的进程执行痕迹，1 001 个攻击的进程执行痕迹，正常的进程执行痕迹中有 69 个为空，可用于实验的共有 2 697 个正常的进程执行痕迹，滑动窗口的大小设置为 9。任意选取其中 200 个正常执行痕迹进行训练。

进程运行是否异常可以从两个方面进行衡量：整个进程的异常度和单个系统调用的异常度。为检测整个进程是否异常，定义整个进程的异常度P。在进程的执行痕迹中，被分类为正常的序列数为S_N，被分类为异常的序列数为S_A，则进程的异常度$P=S_A/(S_N+S_A)$。在每一个系统调用时确定当前系统调用是否正常，可以有效地进行程序行为控制，提高系统的安全防护能力。定义在每一个系统调用处的系统调用异常度为当前系统调用之前 15 个系统调用序列样本中被分类为异常的样本数。

本研究进行了 3 个实验：(1)基于单类 SVM 检测计算进程异常度；(2)基于单类 SVM 检测计算系统调用异常度；(3)基于 RST 修正核单类 SVM 检测计算系统调用异常度。

实验 1 中首先对单类 SVM 进行训练，然后对未知的执行痕迹进行检测，并计算进程异常度。实验 1 的结果如图 2 所示，从中可以看出，利用单类 SVM 对未知执行痕迹进行分类时，异常执行痕迹的进程异常度远高于正常执行痕迹的进程异常度。利用这个特点完全可以将正常执行痕迹与异常执行痕迹区分开。

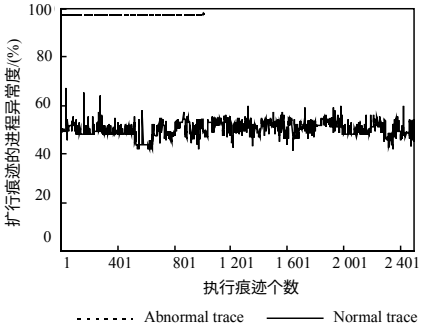

图 2 进程的异常度

实验 2 的目的是观察单类 SVM 检测时的系统调用异常度，从而区分正常系统调用和异常系统调用的可行性。实验 2 的结果如图 3 所示，纵坐标是当前系统调用之前 15 个系统调用序列样本中被分类为异常的样本数。从图 3 中可以看出，仅用单类 SVM 进行分类时，正常痕迹和异常痕迹在一部分系统调用处其系统调用异常度有重叠。因此，单类 SVM 用于检测系统调用异常度需要进行修正。

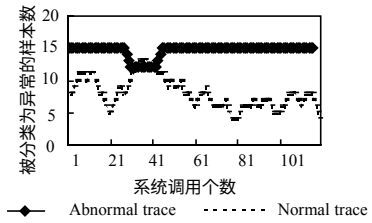

图 3 单类 SVM 分类的系统调用异常度

实验 3 的目的是观察基于 RST 修正核单类 SVM 检测时的系统调用异常度，从而区分正常痕迹和异常痕迹的可行性。实验 3 的结果如图 4 所示。从图 4 中可以看出，使用 RST 修正核单类 SVM 分类，正常痕迹和攻击痕迹的系统调用异常度在所有的系统调用处都有显著的区别，可以用于进行程序行为控制。

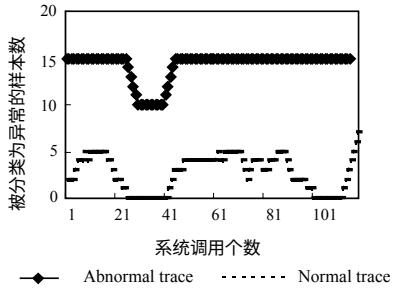

图 4 RST 修正核单类 SVM 分类的系统调用异常度

6 结束语

本文研究并实现了基于粗糙集理论修正核单类支持向量机的程序行为控制系统，提出利用粗糙集理论修正核函数的方法，利用可加载内核模块在内核空间拦截系统调用，并在系统调用的人口点利用基于粗糙集理论修正核单类支持向量机进行实时检测，发现异常则进行实时响应，有效地实现了程序行为控制。通过实验说明，基于 RST 修正核单类 SVM 能够有效地检测程序行为异常。

下一步将对如下几个方面进行研究：(1)正常行为建模的特征提取与特征选择；(2)支持向量机的核函数的比较研究；(3)利用用户空间栈和系统调用参数信息。

参考文献

[1] 蔡忠闽, 管晓宏, 邵 萍, 等. 基于粗糙集理论的入侵检测新方法[J]. 计算机学报, 2003, 26(3): 1-6.

[2] Yao Jingtao, Zhao Songlun, Lisa Fan. An Enhanced Support Vector Machine Model for Intrusion Detection[C]//Proceedings of the International Conference on Rough Sets and Knowledge Technology. Chongqing, China: [s. n.], 2006.

[3] 史忠植. 高级人工智能[M]. 2 版. 北京: 科学出版社, 2006: 306-337.

[4] Scolkopf B, Platt J C, Shawe-Taylor J, et al. Estimating the Support of a High-Dimensional Distribution[J]. Neural Computation, 2001, 13(7): 1443-1471.