# 面向 PCA 异常检测器的毒害攻击和防御机制

钱叶魁[1,2],陈 鸣[1]
(1.解放军理工大学指挥自动化学院,江苏南京 210007; 2.解放军防空兵指挥学院,河南郑州 450052)

**摘 要:** 网络流量异常检测对于保证网络稳定高效运行极为重要.目前基于主成分分析(PCA)的全网络异常检测算法虽然发挥了关键作用,但它还存在着受毒害攻击而失效的问题.为此,深入分析了毒害攻击的机制并对其进行了分类,提出了量化毒害流量的两个测度,并给出了 3 种新的毒害攻击机制;提出了一种基于健壮 PCA 的异常检测算法 RPCA 以抵御毒害攻击.模拟试验结果表明,RPCA 算法在受到多种毒害攻击时仍然具有很好的检测性能,明显优于 PCA 异常检测器,且运行时间能够满足实际网络异常检测的需求.

**关键词:** 异常检测; 毒害攻击; 防御机制; 主成分分析; 健壮性
**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2011) 03-0543-06

# Poison Attack and Defense Strategies on PCA-Based Anomaly Detector

QIAN Ye-kui[1,2], CHEN Ming[1]
(1. *Institute of Command Automation, PLA Univ. of Sci. & Tech., Nanjing, Jiangsu* 210007, *China*;
2. *Air Defence Forces Command Academy of PLA, Zhengzhou, Henan* 450052, *China*)

**Abstract:** Network traffic anomaly detection is crucial to guarantee stable and effective network operation. Nowadays, although PCA-based network-wide anomaly detector plays an important role, it cannot detect anomalous network traffic effectively in face of poison attacks. In order to solve poison attack problem aiming at PCA-based anomaly detector, poison attack strategies are investigated and classified, two metrics for quantifying poison traffic are proposed and three novel poison attack strategies are put forward. A robust PCA-based anomaly detection algorithm (for short RPCA) is proposed to resist poison attacks. Simulation experiment results show that RPCA algorithm can still perform very well in face of poison attacks, obviously superior to PCA-based anomaly detector, and its running time can satisfy the need of practical network anomaly detection.

**Key words:** anomaly detection; poison attacks; defense strategy; principal component analysis; robustness

## 1 引言

基于主成分分析的网络异常检测方法[1](简称 PCA 算法或 PCA 异常检测器)近年来受到了学术界和网络运营商的广泛重视.该方法以流量矩阵作为异常检测的数据源,通过分析流量矩阵高维数据对应的残余向量,实现全网络(network-wide)异常检测.

但是,PCA 异常检测器的健壮性存在着重大缺陷.B.Rubinstein 等人[2]提出了针对 PCA 异常检测器的 4 种毒害攻击机制,并通过模拟试验证实了这些毒害攻击机制能迅速恶化 PCA 异常检测器的检测性能.因此,当前的一个重要研究课题是:深入理解毒害攻击的机制和探讨毒害攻击可能的模式,提出在一定的条件下能够抵御毒害攻击、确保 PCA 异常检测器检测性能的新方法.

本文的主要贡献包括以下三个方面:(1)研究了毒害攻击的机制,给出了针对 PCA 异常检测器的 3 种新毒害攻击模式;第(2)首次将健壮 PCA 应用于网络流量异常检测领域,提出了一种能够抵御毒害攻击的更为健壮的异常检测算法 RPCA;(3)在模拟环境下比较了 PCA 算法和 RPCA 算法的检测性能和检测速度,验证了 RPCA 算法的有效性.

## 2 相关工作

Lakhina 等人提出的基于 PCA 的异常检测算法[1]属于一种全网络异常检测方法,他们首次证实了流量矩阵具有的低维特性以及不同流之间的空间相关性.基于此发现,他们使用 PCA 方法将流量矩阵形成的原始空间分离为正常子空间和异常子空间,并使用 Q 统计量[1]阈值在异常子空间检测网络异常.Ringberg 等人提出了 PCA 异常检测器面临的 4 个挑战[3],其中包括较大的异

收稿日期:2009-10-23;修回日期:2010-10-30
基金项目:国家自然科学基金(No.61070173);国家 863 高技术研究发展计划(No.2007AA01Z418);江苏省自然科学基金(No.BK2009058)

常流量会污染正常子空间,导致检测器无法有效地检测异常.Rubinstein 等人[2]则利用了 PCA 异常检测器的缺陷,提出了 4 种数据毒害机制,并通过试验验证了这些方法能使 PCA 异常检测器的检测性能迅速恶化.

健壮 PCA 是一种改进的 PCA 方法[4~8],作为一种重要的统计和数据分析方法,它已经在化学计量学[9]、生物信息学[10]、计算机视觉[8]等领域得到广泛地应用.

本文首次将健壮 PCA 的思想应用于网络流量异常检测中,提出了一种健壮的网络异常流量检测方法,类似的工作未见报道.

## 3 流量矩阵和 PCA 异常检测器

使用 PCA 异常检测器的前提是获得流量矩阵.本文首先对流量矩阵模型[11]进行简要的描述,然后概述 PCA 异常检测器的工作原理.

### 3.1 流量矩阵

**定义 1** OD 流流量矩阵[11]

OD 流流量矩阵是指一个网络中所有源节点和目的节点对(简称为 OD 对)之间流量需求(traffic demand)的时间序列.

**定义 2** 链路流量矩阵[11]

链路流量矩阵是指网络中所有链路流量大小的时间序列.每条链路的流量是由穿越此链路的 OD 流流量叠加形成的.

**定义 3** 选路矩阵[11]

链路流量和 OD 流流量的关系可以用选路矩阵精确地刻画.假定网络具有 $N$ 条链路和 $P$ 条 OD 流,选路矩阵可表示为 $N\times P$ 的矩阵 $\boldsymbol{A}$,若 OD 流 $j$ 经过链路 $i$,则 $A_{ij}=1$,否则 $A_{ij}=0$.

若 $\boldsymbol{X}$ 表示 $T\times P$ 的 OD 流流量矩阵,$\boldsymbol{Y}$ 表示 $T\times N$ 的链路流量矩阵,则 $\boldsymbol{Y}=\boldsymbol{X}\boldsymbol{A}^{\mathrm{T}}$;$\boldsymbol{X}$ 的第 $t$ 行表示 $t$ 时刻 $p$ 条 OD 流流量测量值构成的向量,简记为 $\boldsymbol{x}(t)=\boldsymbol{X}_{t,\cdot}$;$\boldsymbol{Y}$ 的第 $t$ 行表示 $t$ 时刻 $N$ 条链路流量测量值构成的向量,简记为 $\boldsymbol{y}(t)=\boldsymbol{Y}_{t,\cdot}$.

大部分网络异常行为(如分布式拒绝服务攻击 DDoS、蠕虫扫描、闪拥等)都会引起 OD 流流量的变化,因此,可以将 OD 流流量矩阵作为检测网络异常的数据源.

### 3.2 PCA 异常检测器

从原理上讲,PCA 异常检测器[1]可以分为两个步骤:第一步,将 PCA 应用于 OD 流流量矩阵获得主成分,由此建立正常子空间和异常子空间;第二步,将 OD 流流量测量值向量向异常子空间进行投影,并利用 Q 统计量作为阈值来检测异常.

## 4 毒害攻击机制分析

假定攻击者得知因特网服务提供商(ISP)使用了 PCA 异常检测器,就可能对其发动攻击,使检测器失效.首先对攻击者和检测器作必要的假定,然后定量地分析各种毒害机制.

### 4.1 毒害攻击场景

图 1 给出了具有 4 个 PoP 点的 ISP 网络,在此攻击场景下,假定攻击者的目标是发动一次从 PoP 节点 D 到 PoP 节点 B 的 DoS 攻击而成功地逃避检测.每个 PoP 节点都有入口链路和出口链路,客户流量通过那些入口链路进入 ISP,而通过那些出口链路流出 ISP.在 OD 流路径中与源节点直接相连的那条链路称为源链路,而把与目的节点直接相连的那条链路称为目的链路.

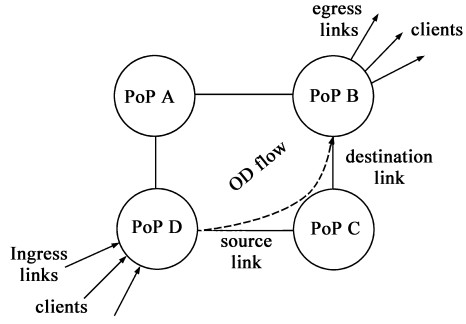


**图1 具有4个PoP点的ISP网络攻击场景**

攻击者为了能够发动针对 PCA 异常检测器的攻击,需要满足如下两点假定[2]:首先攻击者能够沿着他们想要攻击的 OD 流注入毒害流量.在图 1 中,攻击者能够在 PoP 节点 D 经过源链路向 PoP 节点 B 注入毒害流量.其次,假定攻击者能够实时感知源链路的流量大小,以便某些毒害机制能够根据源链路流量大小注入相应强度的毒害流量.

在毒害攻击中,毒害过程持续一周且毒害攻击的强度保持不变的毒害攻击被称为 week-long 攻击.毒害过程持续多周且毒害攻击的强度逐渐增加的毒害攻击被称为 boiling-frog 攻击.

假定 PCA 异常检测器分为两个阶段:训练阶段和检测阶段[6].检测器把前一周的流量矩阵数据集作为训练集,以当前周的流量矩阵数据集作为测试集,根据前一周获得的主成分来检测当前周的流量矩阵数据集中的异常.

### 4.2 毒害机制及其原理

文献[6]提出了 4 种毒害机制.其中,Half Normal 和 Scaled Bernoulli 机制注入的毒害流量大小独立于源链路的背景流量且服从某种概率分布,被称为第 1 类毒害机制;而 Add-Constant-If-Big 和 Add-More-If-Bigger 机制注入的毒害流量大小依赖于源链路的背景流量,被称为第 2 类毒害机制.第 1 类中两种毒害机制之间的区别在于注入的毒害流量大小服从不同的概率分布,而第 2 类中两种机制之间的差异在于注入的毒害流量大小依赖于源

链路的背景流量方式有所不同.

这些机制都具有共同的原理,即攻击者通过向训练集的目标 OD 流注入毒害流量以增加目标 OD 流的方差,从而使得 PCA 异常检测器学习得到的正常子空间向被毒害 OD 流的方向发生歪斜,最终使得出现在测试集中同样目标 OD 流上的异常残余流量减小,提高了异常流量逃避检测的概率.因此,我们可以把这种类型的攻击统称为方差注入攻击,并将毒害流量的方差定义为影响攻击效果的重要测度.

根据上述分析,本文为第 1 类再引入 3 种新的毒害机制:服从 Exponential 分布、Gamma 分布和 Rayleigh 分布的毒害机制,它们具有不同的方差.引入这 3 种新的毒害攻击机制是为了研究毒害攻击流量的大小和方差对毒害攻击效果的影响,当然,还存在其它类似的毒害攻击机制.表 1 列出了 7 种毒害机制及其参数值,每种毒害机制都是在目标 OD 流中注入大小为 $c_t$ 的毒害流量,而 $c_t$ 是以 $\theta$ 为参数的函数,$\theta$ 的取值决定于毒害攻击的强度.

**表 1　7 种毒害机制及其参数值**

| 毒害机制 | 毒害流量的大小 | 毒害流量的均值 | 毒害流量的方差 |
|---|---|---|---|
| Half Normal | $c_t = \lvert n_t \rvert, n_t \sim N(0,\theta^2)$ | $\sqrt{\frac{2}{\pi}}\theta \approx 0.8\theta$ | $(1-\frac{2}{\pi})\theta^2 \approx 0.36\theta^2$ |
| Scaled Bernoulli | $c_t = \theta b_t, b_t \sim b(1,p)$ | $p\theta$ | $p(1-p)\theta^2$ |
| Exponential | $c_t = exprnd(\theta)$ | $\theta$ | $\theta^2$ |
| Gamma | $c_t = gamrnd(1,\theta)$ | $\theta$ | $\theta^2$ |
| Rayleigh | $c_t = raylrnd(\theta)$ | $\sqrt{\pi/2}\theta \approx 1.25\theta$ | $\frac{4-\pi}{2}\theta^2 \approx 0.43\theta^2$ |
| Add-Constant-If-Big | $c_t = \begin{cases}\theta, y_s(t) \geqslant \alpha \\ 0, y_s(t) < \alpha\end{cases}$<br>$mean = \frac{1}{T}\sum_{t=1}^{T} y_s(t)$<br>$\max = \max_{1\leqslant t\leqslant T}(y_s(t))$<br>$mean \leqslant \alpha \leqslant \max$ | 取决于 $y_s(t)$ | 取决于 $y_s(t)$ |
| Add-More-If-Bigger | $c_t = (y_s(t)-\alpha)^\theta$,<br>$\alpha = \frac{1}{T}\sum_{t=1}^{T} y_s(t)$ | 取决于 $y_s(t)$ | 取决于 $y_s(t)$ |

除了毒害流量的方差以外,毒害比率被定义为影响毒害效果的另一个重要测度.所谓毒害比率是指在训练集中注入毒害流量的持续时间与训练集持续时间的比率.一般而言,毒害比率越大则毒害效果越明显.为了研究毒害比率对毒害效果的影响规律,本文还对 Scaled Bernoulli 机制和 Add-Constant-If-Big 机制进行了扩展,分别将这两种机制中的参数设置为可调参数,通过调整该参数可以调节毒害比率.

Add-Constant-If-Big 和 Add-More-If- Bigger 机制需要知道源链路流量大小的均值,根据文献[6],本文假定该均值是相对平稳的,因此可以根据前一周的均值来确定当前周的均值.

## 5　一种防御机制

针对 PCA 异常检测器的攻击机制利用了 PCA 对离群点的敏感性,通过毒害流量使得检测器在训练阶段学习到歪曲的主成分,从而无法有效地检测测试数据中的异常.本文提出一种基于健壮 PCA 的全网络异常检测算法(network-wide anomaly detection algorithm based on robust PCA,简称为 RPCA),该算法在遭受毒害流量攻击的情况下仍然能够学习到最优的主成分,从而有效地检测出异常.

### 5.1　基本思想

ROBPCA[4]是多种健壮 PCA 方法中最著名的一种,它结合了投影追踪(projection pursuit)方法[5]和健壮的协方差估计量(robust covariance estimator)方法[6],能够产生更为健壮的主成分.本文正是基于 ROBPCA 方法的基本思想提出了 RPCA 算法,该算法首先获取 OD 流流量矩阵的健壮主成分,然后利用获得的健壮主成分构建子空间,最后利用 Q 统计量检测异常.

### 5.2　RPCA 算法描述

$T \times P$ 的 OD 流流量矩阵 $\boldsymbol{X}$ 可以看作 $T$ 个 $P$ 维数据点 $\boldsymbol{x}_i$.本文给出 RPCA 算法的核心步骤如下:

**Step 1**　从 $T$ 个数据点中找到 $H$ 个"较少离群的"数据点,然后使用这 $H$ 个数据点的协方差矩阵来获得维度为 $k$ 的初始子空间.

(1)选取 $H = [\alpha T]$,其中 $0.5 < \alpha < 1$,$\alpha$ 的取值决定了方法的健壮性和有效性.当数据集不被离群点污染时,$\alpha$ 越大则获得的子空间越有效,当数据集被离群点污染时,$\alpha$ 越小则获得的子空间越健壮.通常 $\alpha = 0.75$,显然,$1-\alpha$ 表示算法能够抵御的离群点的比率,它正好对应于毒害比率测度.

(2)按式(1)计算所有数据点的离群度(outlyingness),选取 $H$ 个具有最小离群度的数据点,表示为集合 $I_0$.

$$outl_o(\boldsymbol{x}_i) = \max_{v\in B} \frac{\lvert \boldsymbol{x}_i'v - t_{\mathrm{MCD}}(\boldsymbol{x}_j'v)\rvert}{s_{\mathrm{MCD}}(\boldsymbol{x}_j'v)}, i = 1\cdots T \qquad (1)$$

其中 $t_{\mathrm{MCD}}$和 $s_{\mathrm{MCD}}$分别表示一元 Minimum Covariance Determinant (MCD)位置和刻度估计量[12];$B$ 包括通过任意两个数据点的所有方向.

(3) 计算 $I_0$ 中 $H$ 个数据点的均值$\hat{\boldsymbol{\mu}}_0$ 和协方差矩阵$\hat{\boldsymbol{\Sigma}}_0$,即

$$\hat{\boldsymbol{\mu}}_0 = \frac{1}{H}\sum_{i\in I_0} \boldsymbol{x}_i \qquad (2)$$

$$\hat{\boldsymbol{\Sigma}}_0 = \frac{1}{H-1}\sum_{i\in I_0} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_0)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_0)' \qquad (3)$$

(4) 对协方差矩阵 $\hat{\boldsymbol{\Sigma}}_0$ 进行谱分解,即

$$\hat{\boldsymbol{\Sigma}}_0 = \boldsymbol{P}_0 \boldsymbol{L}_0 \boldsymbol{P}'_0 \tag{4}$$

(5) 将所有的数据点向协方差矩阵 $\hat{\boldsymbol{\Sigma}}_0$ 前 $K$ 个特征向量张成的子空间 $V_0$ 进行投影,而 $K$ 可以通过累积方差百分比阈值[1]或健壮 PRESS 算法[13]确定.

**Step 2** 利用 MCD 估计量[6]计算获得健壮的中心和协方差矩阵.

(1) 对于每个数据点,计算它的垂直距离 $OD_i^{(0)}$,即

$$OD_i^{(0)} = \| \boldsymbol{x}_i - \hat{\boldsymbol{x}}_{i,K} \| \tag{5}$$

其中 $\hat{\boldsymbol{x}}_{i,k_0}$ 表示数据点 $x_i$ 在子空间 $V_0$ 中的投影向量.若设置阈值 $c_{OD} = (\hat{\boldsymbol{\mu}} + \hat{\sigma} z_{0.975})^{\frac{3}{2}}$,其中 $\hat{\boldsymbol{\mu}}$ 和 $\hat{\sigma}$ 通过一元 MCD[12]估计得到,$z_{0.975}$ 表示高斯分布的 0.975 分位数,则可以获得所有满足 $OD_i^{(0)} \leqslant c_{OD}$ 的数据点.

(2) 对这些数据点对应的协方差矩阵 $\hat{\boldsymbol{\Sigma}}_1$ 进行谱分解,即

$$\hat{\boldsymbol{\Sigma}}_1 = \boldsymbol{P}_1 \boldsymbol{L}_1 \boldsymbol{P}'_1 \tag{6}$$

(3) 利用前 $K$ 个特征向量张成更为健壮的子空间 $V_1$,并将所有数据点向子空间 $V_1$ 进行投影;然后应用加权 MCD 估计量[6]计算子空间 $V_1$ 中投影数据对应的健壮的中心和健壮的协方差矩阵;最后对健壮的协方差矩阵进行谱分解获得的特征向量就是健壮的主成分.

**Step 3** 利用获得的健壮的主成分构建子空间,然后利用 Q 统计量阈值检测异常.

(1) 在获得 $P$ 个主成分以后,就可以利用前 $K$ 个主成分张成正常子空间 $\hat{S}$,而利用后 $P-K$ 个主成分张成异常子空间 $\tilde{S}$.

(2) 任意时刻流量测量值的向量 $\boldsymbol{x}$ 就可以向异常子空间进行投影,$\tilde{\boldsymbol{x}}$ 表示 $\boldsymbol{x}$ 在异常子空间 $\tilde{S}$ 中的投影向量,称为残余流量(residual traffic).将前 $K$ 个主成分排列成 $P \times K$ 的矩阵 $\boldsymbol{P}_{P,K}$,则

$$\tilde{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{P}_{P,K} \boldsymbol{P}_{P,K}^{\mathrm{T}}) x = \tilde{\boldsymbol{C}} \boldsymbol{x} \tag{7}$$

流量大小异常通常会引起 $\tilde{x}$ 发生较大的变化,因此可以通过设置 $Q$ 统计量阈值 $Q_\beta$ 来检测异常,若 $\| \tilde{x} \|^2 > Q_\beta$,则判定为异常,否则为正常.

## 6 试验评价

为了客观地评价毒害攻击机制对 PCA 异常检测器的影响效果以及 RPCA 算法在遭受毒害攻击时的健壮程度,本文设计一系列仿真试验并对试验结果进行深入的分析.

### 6.1 试验方法

本文同时采用 Abilene 网络流量矩阵[1]和 GEANT 网络流量矩阵[14]作为试验数据集.

考虑到网络流量通常由 3 种成分构成[1]:近似周期性的正常成分、高斯噪声成分和异常成分.因此,本文选择 Abilene 网络第 1 周流量矩阵作为原始数据,按照下列步骤人工合成不含异常的基准流量矩阵.

(1)本文利用 7 天、5 天、3 天、24 小时、12 小时等多种不同周期的傅立叶基函数以及常数项的线性组合来逼近原始数据的每条 OD 流流量,其中某条 OD 流流量大小的时间序列.

(2)在第一步产生的流量矩阵的每条 OD 流流量上加入零均值的高斯噪声流量,获得不含异常的基准流量矩阵,其中某条 OD 流流量大小的时间序列.

在训练阶段,为了模拟 week-long 攻击,本文把基准流量矩阵当作训练集,依次选择每个 OD 流当作目标 OD 流,按照表 1 中 7 种不同机制分别注入毒害流量.为了模拟 boiling-frog 攻击,我们产生多周的基准流量矩阵当作训练集,在每周的基准流量矩阵中采用与 week-long 攻击类似的方法注入毒害流量,每周依次增加毒害攻击的强度.

在检测阶段,将基准流量矩阵当作测试集,在同样的 OD 流的所有时间间隔内顺序注入攻击流量,攻击流量大小取 $1.5 \times 8 \times 10^7$[1].本文采用漏报率(false negative ratio, FNR)作为评价检测算法的性能测度.需要指出的是,评价异常检测算法的检测性能通常需要同时考虑漏报率和误报率,但是由于本试验中,在选择置信区间为 0.95 的 Q 统计量作为阈值时,算法几乎不会出现误报的现象,因此本文仅仅考察该算法的漏报率.由于除 Add-More-If-Bigger 机制以外,其余 6 种毒害攻击注入的流量大小服从概率分布,具有随机性,因此对于这 6 种毒害攻击,本文重复 10 次,然后求取漏报率的平均值.

### 6.2 试验结果分析

对于 week-long 攻击,分别采用表 1 中 7 种不同的毒害攻击机制,不断增大毒害攻击的强度,计算 PCA 和 RPCA 算法的漏报率,限于篇幅,仅仅显示部分结果如图 2 所示.在 Exponential 和 Add-Constant-If-Big 机制情形下,当 PCA 算法的漏报率达到 100%时,RPCA 算法的漏报率几乎为 0%,其中 Add-Constant-If-Big 机制的参数 $\alpha = mean + (\max - mean)/2$;从整体趋势上看,随着毒害强度的增加,PCA 算法的漏报率呈现出显著的上升趋势,而 RPCA 算法的漏报率上升缓慢得多或者不变.因此在各种毒害攻击情形下,RPCA 算法的漏报率都远远低于 PCA 算法,RPCA 算法在遭受毒害攻击时表现出更为健壮的检测性能,能够更好地抵御 week-long 攻击.

对于 boiling-frog 攻击,分别采用 7 种不同的毒害攻击机制,毒害的持续时间为 1 ~ 20 周,相邻周之间毒害强度的增长率 $g \in \{1\%, 3\%\}$,不断延长毒害攻击的持续时间(以周为单位),计算 PCA 和 RPCA 算法的漏报率,限于篇幅,仅仅显示部分结果如图 3 所示.随着毒害持续时

间的延长,PCA 算法的漏报率呈现出显著的增长趋势,而 RPCA 算法的漏报率上升缓慢得多,且 RPCA 算法的漏报率一直远小于 PCA 算法.因此,RPCA 算法比 PCA 算法具有更强的健壮性,能够有效地抵御 boiling-frog 攻击.

为了评价毒害流量的方差对 PCA 算法和 RPCA 算法检测性能的影响,本文以 4 种不同的毒害机制为例,不断增大毒害攻击的强度,PCA 算法和 RPCA 算法的漏报率变化曲线见图 4.对于 PCA 算法,Scaled Bernoulli 机制对应的漏报率最小,Rayleigh 机制对应的漏报率次之,而 Exponential 机制和 Gamma 机制对应的漏报率最大且两者非常接近.可见,毒害机制注入的毒害流量方差越大则 PCA 算法的检测性能越差,方差相同则 PCA 算法的检测性能非常接近.同样地,对于 RPCA 算法上述规律也成立,只是对于同样的毒害机制,RPCA 算法的检测性能远远优于 PCA 算法.

为了评价毒害比率对 PCA 算法和 RPCA 算法检测性能的影响,本文以 Scaled Bernoulli 机制为例进行讨论.为 Scaled Bernoulli 机制的参数设置两个不同的数值以代表不同的毒害比率,不断增大毒害攻击的强度,PCA 算法和 RPCA 算法的漏报率变化曲线见图 5.设置 RPCA 算法的参数 $\alpha = 0.75$,设置 Scaled Bernoulli 机制的参数 $p = 0.25$ 或 0.30.可见,毒害比率越大,PCA 算法和 RPCA 算法的漏报率越大.但是,对于 RPCA 算法,当毒害比率未超过其能够抵御的限度时具有很好的检测性能,而一旦超过该限度则检测性能将迅速恶化.

算法的运行时间也是评价算法的重要指标.图 6 给出了执行 PCA 算法和 RPCA 算法时随着样本数增加的 CPU 运行时间,其中计算机的 CPU 主频为 2.33GHz,内存为 2GB,选取维数为 144 的 Abilene 流量矩阵以及维数为 529 的 GEANT 流量矩阵.可见,随着样本数的增加,两个算法的 CPU 运行时间呈近似线性增加;而对于同样的样本数,维数越高,两种算法的 CPU 运行时间越长;RPCA 算法的检测速度比 PCA 算法慢,但能够满足实际检测需要.

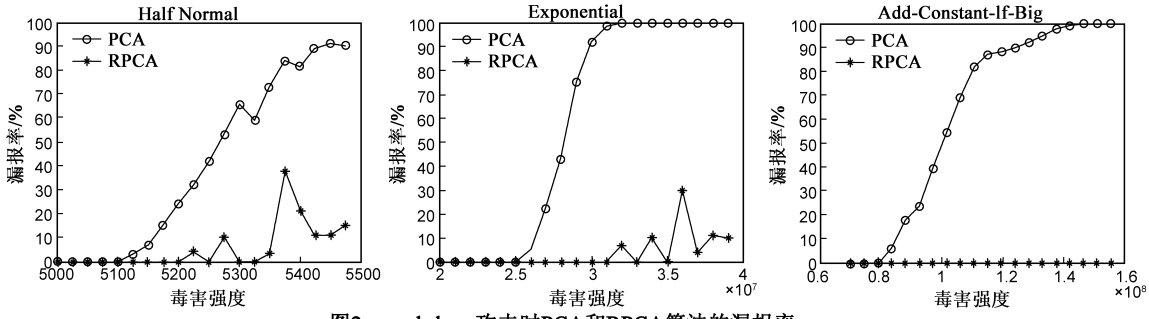


图2　week-long攻击时PCA和RPCA算法的漏报率

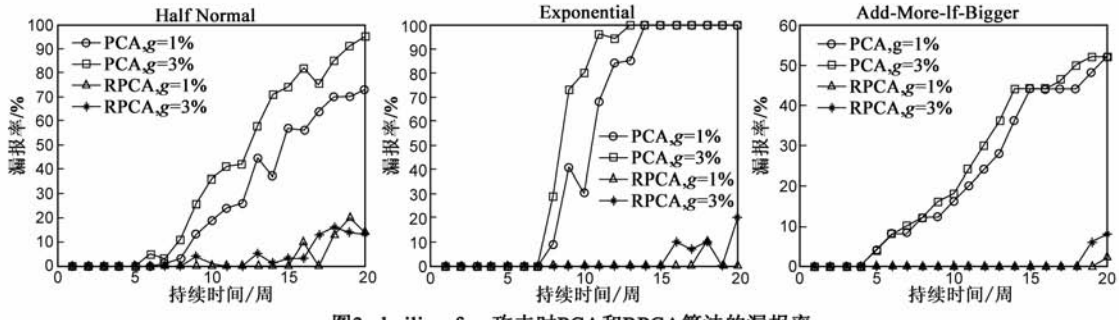


图3　boiling-frog攻击时PCA和RPCA算法的漏报率

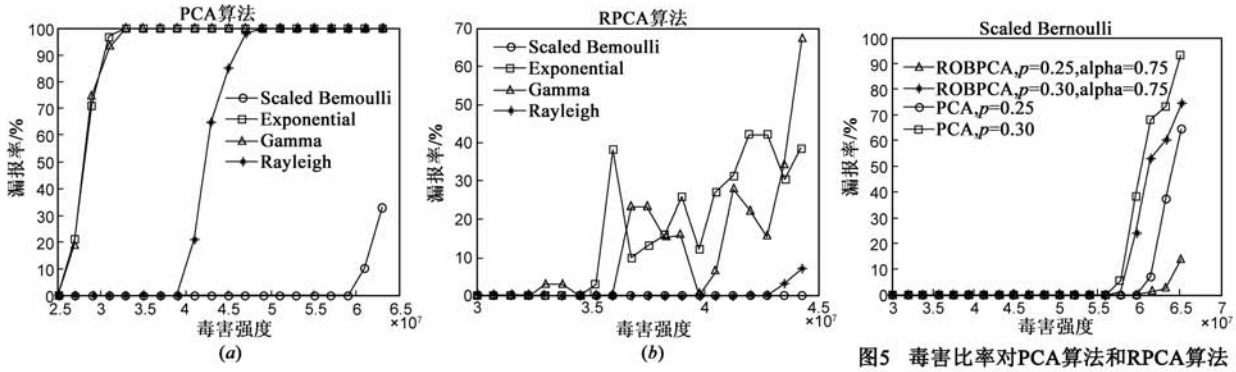


图4　4种毒害攻击时PCA算法和RPCA算法的漏报率

图5　毒害比率对PCA算法和RPCA算法漏报率的影响

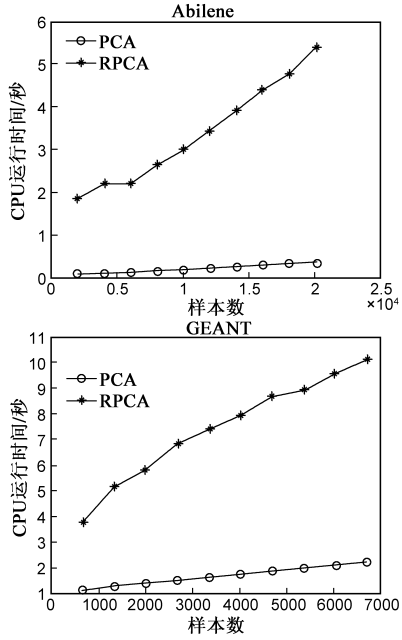


图6 PCA算法和RPCA算法的CPU运行时间

## 7 结论

针对 PCA 异常检测器在毒害攻击下无法检测网络异常流量的问题,本文对毒害攻击机制进行了研究,对毒害攻击进行了分类且提出了毒害流量的两个测度,并给出了 3 种新毒害攻击机制;提出了一种基于健壮 PCA 的异常检测算法 RPCA.模拟试验结果表明,RPCA 算法在遭受毒害攻击时仍然具有很好的检测效果,明显优于 PCA 异常检测器,且运行时间能够满足网络实际检测和管理的需求.下一步我们将研究在面临不完整流量矩阵的情况下,如何保证 PCA 异常检测器具有健壮的检测性能.

### 参考文献

[1] Lakhina A, Crovella M, Diot C. Diagnosing network-wide traffic anomalies[A]. ACM SIGCOMM[C]. Portland, Oregon, USA, 2004. 123 – 134.

[2] Rubinstein B I P, Nelson B, Huang L, et al. Compromising PCA-based Anomaly Detectors for Network-wide Traffic[R]. Technical Report UCB/EECS-2008-73, 2009.

[3] Ringberg H, Soule A, Rexford J, et al. Sensitivity of PCA for traffic anomaly detection[A]. SIGMETRICS[C]. San Diego, California, USA, 2007. 212 – 223.

[4] Hubert M, Rousseeuw P J, Branden K V. ROBPCA: a new approach to robust principal component analysis[J]. Technometrics, 2005, 47(3): 64 – 79.

[5] Hubert M, Rousseeuw P J, Verboven. A fast robust method for principal components with applications to chemometrics[J]. Chemometrics and Intelligent Laboratory Systems, 2002, 60(3): 101 – 111.

[6] Rousseeuw P J, Van Driessen K. A fast algorithm for the minimum covariance determinant estimator[J]. Technometrics, 1999, 41(5): 212 – 223.

[7] Maronna R. Principal components and orthogonal regression based on robust scales[J]. Technometrics, 2005, 47(6): 264 – 273.

[8] Torre F D L, Black M J. Robust principal component analysis for computer vision[A]. ICCV[C]. Vancouver, 2001. 321 – 330.

[9] Debruyne M, Engelen S, Hubert M, et al. Robustness and outlier detection in chemometrics[J]. Critical Reviews in Analytical Chemistry, 2006, 36(6): 245 – 256.

[10] Hubert M, Engelen S. Robust PCA and classification in biosciences[J]. Bioinformatics, 2004, 20(2): 1728 – 1736.

[11] Vardi V. Network tomography: estimating source-destination traffic intensities from link data[J]. Journal of the American Statistical Association, 1996, 91(3): 365 – 377.

[12] Rousseeuw P J. Least median of squares regression[J]. Journal of the American Statistical Association, 1984, 79(6): 871 – 880.

[13] Hubert M, Engelen S. Fast cross-validation of high-breakdown resampling methods for PCA[J]. Computational Statistics and Data Analysis, 2007, 51(3): 5013 – 5024.

[14] Uhlig S, Quoitin B, Lepropre J, et al. Providing public intradomain traffic matrices to the research community[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(1): 231 – 242.

### 作者简介

**钱叶魁** 男,1980 年生于安徽安庆.解放军理工大学博士.研究方向为网络测量和网络安全.

E-mail: qyk1129@hotmail.com

**陈 鸣** 男,1956 年生于江苏无锡.博士,解放军理工大学教授,博士生导师,研究方向为网络测量和网络管理等.