

一种基于主成分分析算法的网络异常检测实现

付 强, 甘 亮, 李爱平, 吴泉源

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

[摘要] 针对分布式网络的网络异常检测, 提出一种多维数据特征自适应的异常检测算法, 算法在主成分分析算法 (PCA) 的基础上进行异常特征自适应修正. 在对网络流量数据经过了 PCA 处理后, 确定贡献率高的维度, 给出异常与维度特征的关联, 进行特征自适应修正. 实验结果表明, 算法降低了网络异常检测的执行开销, 提高了网络异常检测的报警精度.

[关键词] 网络异常, PCA, 自适应算法

[中图分类号] TP 393 [文献标识码] A [文章编号] 1672-1292(2008)04-0013-04

Implementation of Adapting Algorithm of Anomalies Detection Based on PCA

Fu Qiang Gan Liang Li Aiping Wu Quanyuan

(School of Computer Science, National University of Defense Technology Changsha 410073, China)

Abstract Aiming at the network anomaly detection of distributed networks, the paper suggests a multidimensional adapting algorithm of anomalies detection. The algorithm, based on PCA, can change itself automatically for new anomaly. Having processed network flow data by PCA algorithms, we can get the dimensions which contribute most and give the relation between anomalies and dimensions, and make self-adaption of features. The experimental results show that the algorithm reduces the cost of network anomaly detection, and improves the alert precision of network anomaly detection.

Key words network anomaly, PCA, adapting algorithm

网络流量异常检测在网络安全监控中具有重要意义, 它可以对异常事件有可能给网络造成拥塞以及网络安全攻击 (如 DDOS) 起到提前预测的作用, 从而降低甚至避免损失. 本文主要采用主成分分析 (PCA) 算法进行预处理, 后期对数据进行自适应调整. 在时间层面上考虑高效性, 通过对网络流量检测中的复杂特征项进行降维处理, 通过自适应算法, 剔除对检测结果贡献率小的特征项, 直接定位在影响检测的主特征项上, 避免了计算冗余, 提高了检测的后续处理速度. 在空间层面上通过增加特征表项来提高检测效果, 根据异常特征对流量数据进行分流处理, 通过增加异常分类数据表, 针对特定的网络异常实施有针对性的处理, 提高了对大量数据的检索速度和检测命中率, 用空间上的开销获得了速度性能上的提高.

网络流量数据通过 PCA 处理后, 通过自适应的递归迭代算法对流量数据进行深度检测, 准确定位异常种类, 并且对未知异常给出警告处理, 建立新的类别异常, 使其成为后续检测新的异常类别区分方法.

1 相关研究

在流量异常检测处理方面, 国内外的很多学者和相关研究人员提出了很多检测方法. 比较经典的流量检测方法是基于阈值的检测方法, 这种方法通过对历史数据的分析建立正常的参考基线范围, 一旦超出此范围就判断为异常. 基于统计的检测, 如一般似然比 (GLR) 检测方法, 它考虑两个相邻的时间窗口以及将这两个窗口合并之后的窗口, 对这些窗口采用自回归模型拟合, 并计算各窗口序列残差的联合似然比, 然后与某个预先设定的阈值进行比较, 当超过阈值时, 则窗口边界被认定为异常点^[1]. 基于变换域的方法通常将时域的流量信号变换到频域或者小波域, 然后依据变换后的空间特征进行异常检测. Barford P 等人将小波分析理论运用于流量异常检测^[1], 并给出了基于其理论的 4 类异常结果, 但是该方法复杂性太高, 时间耗费又太大, 不适合应用于大规模的网络流量. Lakhina 等人将源和目标之间的数据流高维结构空间进

收稿日期: 2008-06-18
基金项目: 国家“863”计划 (2007AA01Z474, 2006AA01Z451 和 2007AA010502) 资助项目.
通讯联系人: 吴泉源, 教授, 博士生导师, 研究方向: 人工智能和网络安全. E-mail: quanyuan_wu@live.cn

行 PCA 降维分解^[2], 归结到主成分上, 重构网络流的特征, 并以此发展出一套检测方法. 基于 Markov 模型的网络状态转换概率检测方法, 将每种类型的事件定义为系统状态^[3], 通过过程转换模型来描述所预测的正常的网络特征, 当到来的流量特征与期望特征产生偏差时进行报警.

上述算法在应用到 Netflow 上的大规模流量检测上都有一些弊端和不足. 本文集成了上述某些算法的优点, 并在实际应用中对其进行了适应性的改进, 使得时间效率、速度和检测准确性得到了相应提高.

2 PCA 的自适应算法

2.1 PCA 数据降维的原理

设 $x = (x_1, x_2 \dots x_p)$ 是 P 维随机变量, U 是正交矩阵, 使得 $y = Ux$ 具有 $\text{cov}y = U \text{cov}x U^T = \Lambda$ 的协方差阵, 则称 y_i 是 (关于 x 的) 第 i 个主成分, $i = 1 \dots, p$, 并称 $\frac{\lambda_i}{\sum_{j=1}^p \lambda_j}$ 是第 i 个主成分贡献率. $\frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^p \lambda_j}$ 是前 k 个主

成分的累计贡献率, 其中 Λ 是对角阵 $\text{diag}(\lambda_1, \lambda_2 \dots \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$ 这里 λ_j 是 $\text{cov}x$ 的特征值. 现记 $U = (u_1 u_2 \dots u_p)$, 则 u_i 正是 $\text{cov}x$ 的相应于 λ_i 的特征向量, 如特征值中有一部分为零 (设为 $\lambda_{r+1} = \lambda_{r+2} = \dots = \lambda_p = 0$), 这意味着 y_{r+1}, \dots, y_p 可不必在统计中给予评论, 但是 $\text{cov}x$ 不一定退化, 即 $\lambda_p > 0$ 我们不会应用 p 个主成分, 主成分的目的是为了减少变量的个数, 所以一般用 $m < p$ 个主成分, m 一般是 $\frac{\sum_{j=1}^m \lambda_j}{\sum_{j=1}^p \lambda_j} \geq$

α 就可. 由于 $y_i = u_i \cdot x$, 所以一个主成分对应着 $\text{cov}x$ 的一个特征向量, 故在实际中主要求出 $\text{cov}x$ 的特征向量及特征值并且按照 α 的值进行筛选, 就可以得到主成分.

2.2 自适应类别划分检测方法

自适应类别划分方法对于以往经验取得的网络异常特征进行细化, 获得相关异常的经验特征, 并以此对网络流量分类, 将分类情况作为数据检测的首选. 因网络上的异常大部分都是已经过确定入库的异常事件 (如表 1), 当按以往经验建立的分类标准无法将某个异常点归于某类时, 就需要自适应地去建立新的异常类别. 此时, 经过维度的缩减, 确定了某几个数据特征确实不能吻合以往类别而又非正常时, 将自适应地添加新的异常情况入经验库, 并给出分析概率, 当建立起新的特征库后, 就可增加比对检测的程序.

表 1 网络流量异常分类

Table 1 Anomaly of netflow traffic

异常	定义	特征	示例
Dos DDos	对一台受害主机的 (分布式) 拒绝服务攻击	单个目的 IP, 源 IP 无特殊性, 一般不超过 20 min	多台主机同时向单个目的 IP 的易受 Dos 攻击端口发送大量数据
FLASH CROWD	对资源或服务非正常的大量需求	在单个目的 IP, 特殊端口的数据包, 持续时间短	多台主机对单个 IP 的大量 web 请求 (80 端口)
SCAN	扫描主机脆弱性端口 扫描网络中的目标端口	单个源 IP, 目的 IP 和端口无特殊性, 低于 10 min	对 139 (NetBDS) 端口的网络扫描
WORM	利用安全缺陷, 自繁殖代码在网络中传播	对 IP 地址无特殊性, 有特殊的端口	1433 端口流量 (MSQL-Snake 蠕虫)

对于正常的网络数据流, 有 $\hat{y}_{n+1} = ay_n + a(1-a)y_{n-1} + a(1-a)^2y_{n-2}$ 且 $\hat{y}_n = \sum_{i=0}^{\infty} w_i y_{n-i}$ w_i 为权值, 累加后为 1, $w_i = a^i (1-a)^{i-1}$. $\hat{y}_{n+1} = ay_n + a(1-a)y_{n-1} + a(1-a)^2y_{n-2}$, $\hat{y}_{n+1} = ay_n + (1-a)\hat{y}_n$ 其中 \hat{y}_{n+1} 是下一个预测的值, y_n 是当前的观测值, \hat{y}_n 是前一次检测的预测值, \hat{y}_n 为上一次检测后得到的确定值. a 是一个平滑整个迭代过程的参数, 经过之前 1 h 的数据检测, 建立合适的 a 值, 初始设定为 1, 每次修正增加或者减少 0.1 单位. 对于 \hat{y}_n , 变换其不同的特征项, 获取不同的异常检测迭代等式, 当出现检测结果与预测结果发生偏差时, 对网络流的数据进一步深度匹配检测, 如果在匹配时没有相应的特征项所带来的异常定义, 则提示用户对其进行分析确立新的异常点. 系统实现中的算法如表 2 所示.

表 2 算法伪码实现
Table 2 Realization of the algorithm

算法名称: 数据降维检测算法
输入: DataArray1[]; //存储原始网络流量数据 DataArray2[]; //存储要关注的每种异常的特征 AnomalyFlag[]; //异常标示分类表 α ; //权重
输出: 异常分类表, 异常检测结果
过程: GetCurrentTime(); While(each minute in recent 1hour) { Read netflow_data from DataBase; Store in DataArray1[]; while(AnomalyFlag[i] is not null i++) {PCA(α , DataArray1, AnomalyFlag, DataArray2); Detect(AnomalyFlag, DataArray2, α);} Function PCA(arg1, arg2, arg3, arg4) {DataArray2= Matrix(DataArray1); //对原始数据进行处理得到最后的相关矩阵, 按照选取 m 值的标准对原始数据特征进行筛选, 放入 DataArray2中去, 同时确定 α 是否为调整值的增量 } Function Detect(arg1, arg2, arg3) { 按照 AnomalyFlag 的异常定义对 DataArray2 进行检测, 相当于 DataArray2 流经 AnomalyFlag 规定的异常槽, 当出现波动时, 修改 α 的增量和对 PCA 的回馈并且将预测值与实际进行比较, 从而修改检测的波动区间 } }

系统的整体网络流检测框架如图 1 所示。

3 基于 PCA 的自适应算法在网络异常检测中的实现

实际测试中采用了 Netflow 中的数据进行了实验和测试, 数据的采集是采用思科的网上数据采集设备, 原始数据主要有以下特征: 源 IP、目的 IP、源端口、目的端口、协议号、字节长度、包数据量; 其中源 IP 是发起连接的机器地址或者序列号, 目的 IP 是报文要到达的机器地址, 源端口和目的端口分别指源主机和目标主机开启的端口, 协议号和字节长度以及包数量都是报文方面的信息. 对于上述的数据, 又抽离出了如下的多维向量, 分别按照单位时间内发生的流量中的源 IP 数量、目的 IP 数量、每种协议的发生次数、源端口数量、目的端口数量、包数据量、字节数据量、TOP1 源 IP、TOP1 目的 IP、TOP1 协议、TOP1 源端口、TOP1 目的端口等一系列待选维度, 通过每一分钟取样一次, 跨度 60min 的数据对网络流量异常情况进行检测。

设定矩阵模型 Y , 利用产生向量维度函数在候选维度集合里选择 m 个候选维, 置初始修正参数 m 为 k , 循环提取数据项各列, 随机抽取 k 个作为候选维度, 并标记每个数据项的可用性, 读取对应于候选维度的相应数据, 并添加到 Y 矩阵, 对 Y 矩阵应用 PCA 算法进行处理. 对贡献率小于 α 的维进行剔除, 对贡献率大于 r 的维度, 增加 m 值, 将最新的 m 值传给维度控制函数, 作为维度控制函数下一次获取维度的依据, 对于贡献率大于 α 、小于 r 的维度, 作为衡量网络出现某种异常的标准. 对经过 PCA 处理的矩阵 Y 进行分类划分, 划分正常与异常类别, 对取得的这几个贡献率大的维度取均值与类别尺度进行比对, 当某一分钟内的均值与以往的差超过一定阈值的时候就把它作为异常处理, 并且确定了为何种异常, 如图 2 所示. 当通过自适应检测算法进行处理后, 根据关注点的不同, 会有不同的异常检测结果。

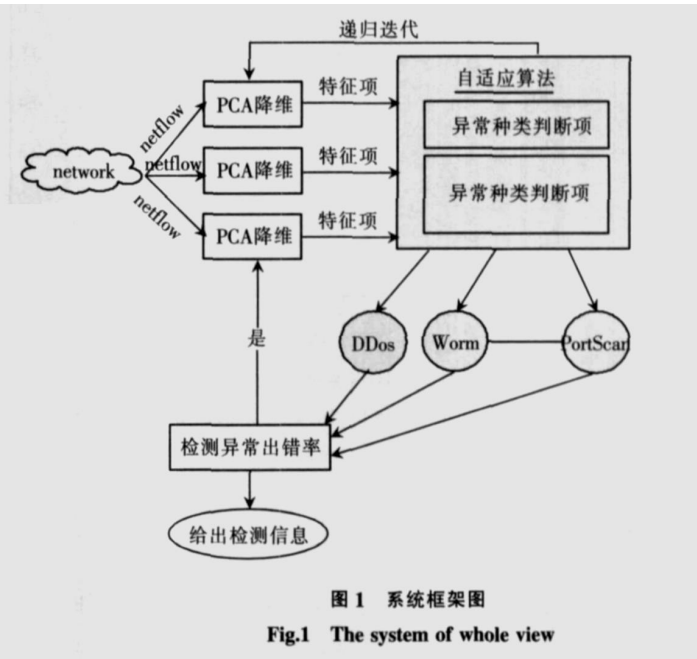


图 1 系统框架图
Fig.1 The system of whole view

设定矩阵模型 Y , 利用产生向量维度函数在候选维度集合里选择 m 个候选维, 置初始修正参数 m 为 k , 循环提取数据项各列, 随机抽取 k 个作为候选维度, 并标记每个数据项的可用性, 读取对应于候选维度的相应数据, 并添加到 Y 矩阵, 对 Y 矩阵应用 PCA 算法进行处理. 对贡献率小于 α 的维进行剔除, 对贡献率大于 r 的维度, 增加 m 值, 将最新的 m 值传给维度控制函数, 作为维度控制函数下一次获取维度的依据, 对于贡献率大于 α 、小于 r 的维度, 作为衡量网络出现某种异常的标准. 对经过 PCA 处理的矩阵 Y 进行分类划分, 划分正常与异常类别, 对取得的这几个贡献率大的维度取均值与类别尺度进行比对, 当某一分钟内的均值与以往的差超过一定阈值的时候就把它作为异常处理, 并且确定了为何种异常, 如图 2 所示. 当通过自适应检测算法进行处理后, 根据关注点的不同, 会有不同的异常检测结果。

4 对实验结果的分析

经过在 Netflow 实际数据上的检测,发现经过降维后的数据和直接检测在速度上的提高.在同样的机器配置条件下,流量相同情况下处理速度的对比见图 3 通过自适应的检测算法后,自行调整阈值,在异常的检测上的效果见图 4.

图 3 结果展示了流量发生变化后,处理相关数据流所耗费的时间情况.而未经处理的数据流,在没有流量增加大的情况下,与降维处理的速度有了差别.图 4 结果中,阈值线 1 之间对应的是异常情况 1,可以看到,异常数累计是增加的,这样阈值不能覆盖 60% 异常情况,会带来预警的丢失;阈值线 2 对应的异常情况 2 可以看到,经过处理,大部分异常都落在了阈值 α 之间,对于异常能够达到覆盖率 50% 以上或不低于此指标.

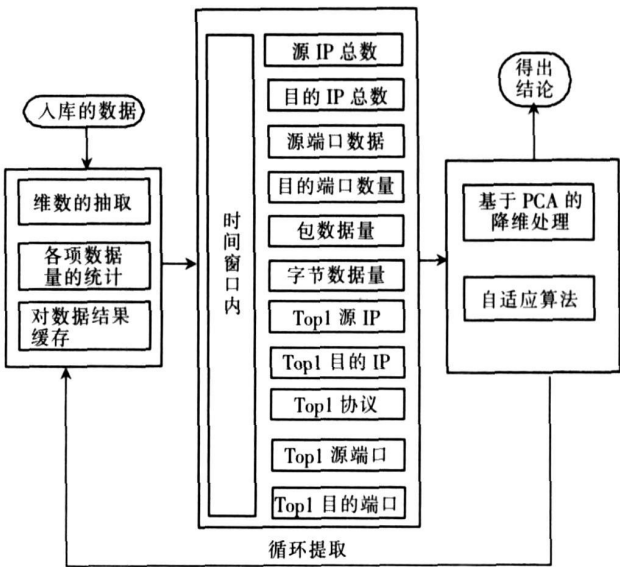


图 2 Netflow 数据
Fig.2 The data of Netflow

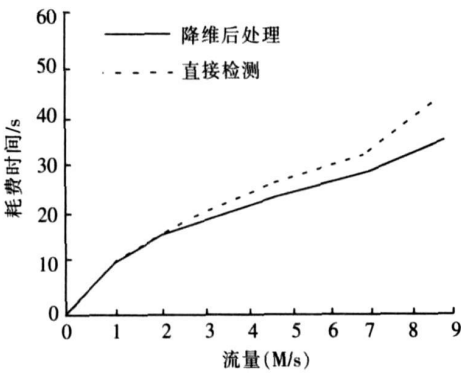


图 3 相同流量处理时间比较
Fig.3 Time used by the same netflow traffic

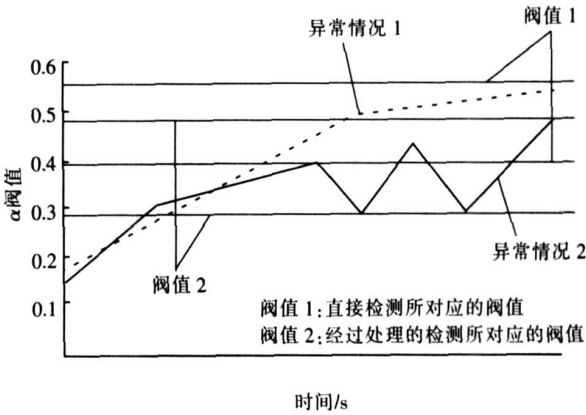


图 4 同网络情况下阈值效果比较
Fig.4 Threshold value of the same network

5 结语

试验结果说明,在对大流量数据进行降维处理后,能够很好地提高检测速度,在对网络数据流进行自适应的异常检测后,检测效果得到了改进,实现起来也容易些.但是,因为自适应是建立在特征维度匹配上的,需要建立特征经验组,而且对于没有关注过的异常,需要进行长时间的检测来确定适应的阈值.

[参考文献] (References)

[1] Huang Ling,Nguye Xuan bong,Minos Garofalakis,et al. Communication-efficient on line detection of network-wide anomalies[C] // Proceedings of the 26th IEEE International Conference on Computer Communications. Anchorage, AK: IEEE Computer Society Press, 2007: 134-142.

[2] Li Xiaokui,Han Jiawei. Mining approximate Top-K subspace anomalies in multidimensional time series data[C] // Proceedings of the 33rd International Conference on Very Large Data Bases. Vienna, Austria: VLDB Endowment, 2007: 447-458.

[3] Li Xin,Bian Fang,Crovella Mark,et al. Detection and identification of network anomalies using sketch subspaces[C] // Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement. New York, USA: ACM, 2006: 147-152.

[责任编辑: 严海琳]