Computer Science and Application 计算机科学与应用, 2018, 8(5), 829-839
Published Online May 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.85092

Hans汉斯

A Design and Simulation of Adaptive Intrusion Detection System Based on Data Mining

Xin Su, Tingjun Shi

Yangzhou University, Yangzhou Jiangsu
Email: boysuxixi@126.com

Received: May 12th, 2018; accepted: May 24th, 2018; published: May 30th, 2018

Abstract

Rule matching detection today dose not adapt to the rapid update of network and system and it is easy to get unknown attack. Considering the weakness of modern rule matching detection technique, the article supposes to apply adaptive intrusion detection through adaptive strategies. According to the technique above, the article proposes to select and adjust detection strategies according to the description of the system environment and the safety threshold. For cases of being less than or equal to safety threshold value, the intrusion detection method based on spectral clustering is adopted, with the aim of maximizing internal weights of subgraphs. This approach is based on division method in graph theory. Besides, weights between subgraphs should be minimized in cluster process. Normal/abnormal behavior libraries were divided according to the fact that the number of normal behavior was far greater than that of abnormal behavior. For cases of being more than safety threshold value, using the isolated forest detection method, this method can find data that does not conform well with other data laws in a large number of data, and thus find abnormal data to achieve the detection effect. Experiments show that this method can reduce the false alarm rate and improve efficiency.

Keywords

Data Mining, Clustering, Isolates Forest, Intrusion Detection, Adaptation

基于数据挖据的自适应入侵检测系统设计与仿真

苏　昕，史庭俊

扬州大学，江苏 扬州
Email: boysuxixi@126.com

文章引用: 苏昕, 史庭俊. 基于数据挖据的自适应入侵检测系统设计与仿真[J]. 计算机科学与应用, 2018, 8(5): 829-839. DOI: 10.12677/csa.2018.85092

收稿日期：2018年5月12日；录用日期：2018年5月24日；发布日期：2018年5月30日

摘　要

如今的规则匹配检测不能适应网络与系统更新的速度，容易受到未知的攻击。针对这一问题，本文通过使用自适应策略来实现入侵检测。该入侵检测方法首先由系统环境描述和安全阈值来选择检测策略，对于小于或等于安全阈值的情况，采用基于谱聚类的入侵检测方法，该方法基于采用图论的划分方法使得划分后的子图内部(同一个类)权值和尽量大，不同子图(不同类)之间权重和尽量小实现聚类，根据正常行为数量远大于异常行为数量划分出分正常/异常行为库；对于大于安全阈值的情况，采用孤立森林检测方法，该方法可以在找到大多数数据规律不符合的数据，从而找到异常数据达到检测效果。通过实验表明，该方法能够降低误报率且能够提高效率。

关键词

数据挖掘，聚类，孤立森林，入侵检测，自适应

1. 引言

随着网络技术的不断发展和网络应用范围的不断扩大，对网络的各类攻击与破坏也与日俱增。由于网络攻击手段的多元化、复杂化和智能化，单纯依赖防火墙等静态防御已难以胜任网络安全的需要。入侵检测是保证网络和系统安全的重要手段，来避免由入侵者(如攻击者和黑客)通过一些技术手段对网络和系统来进行的非法操作。随着传输数据爆炸式增长和高速网络广泛运用，传统的入侵检测方法已经过时并且不能满足目前的检测要求。此外，当一个入侵检测系统工作的时候不应该影响网络系统的正常运行，特别是在大数据处理和高速网络环境。由于现在的网络和系统环境复杂，入侵检测系统不仅能有效的检测入侵而且还需要具有自适应机制。针对这个问题，本文给出了一种自适应选择入侵检测策略的方法并给出相应的模型。模型中使用了孤立森林算法和谱聚类算法。

本文组织如下：第一节为相关工作，第二节介绍了模型结构及自适应策略，第三节和第四节主要描述了孤立算法和谱聚类算法，第五节为数据处理，第六节主要是实验和仿真，最后一节为总结和未来展望。

2. 相关工作

关于入侵检测的研究，文献[1]阐述了一个入侵检测系统(IDS)的通用模型，它将一个入侵检测系统分为事件产生器、事件分析器、响应单元、事件数据库。当前的入侵检测概述技术和相关问题中提出[2]，主要的有基于分类的方法是将所有流量分类为正常或恶意来减少误报和漏报报警。例如，通过规则匹配如源 IP 地址、目标 IP 地址、源和目标的端口号等来形成入侵警报[3] [4]，缺点是漏报率高。另一类指的是通过管理员及专家系统判断[5] [6]。这个方法主要优点是降低了误报率，但是需要专家知识来判断，这个过程消耗时间过长。近年来数据挖掘和机器学习方法已被用于入侵，来改善现有系统的性能[7] [8] [9]，利用这些机器学习算法来分类入侵警报，例如使用聚类解决当前的问题。文献[10]首先确定两个最佳的初

始聚类中心然后利用最大、最小距离的层次聚类算法和 DBI 指标确定剩余初始聚类中心，解决了 k-means 算法初始 K 个中心不容易确定，容易导致局部最优的问题。这些研究大都通过改进聚类算法增强检测效果，一方面增加了算法的复杂度，另一方面这些算法存在参数确定繁琐且不能保证最优和动态改变。在自适应安全方面，文献[11]提出了自适应安全机制。自适应安全是指系统采用非固定的安全工作模式，为获得最佳服务性能，能够根据其状态或环境的变化动态调整运行行为和安全规则。文献[12]采用哈休控制图来确定系统环境的安全状态，但是只选择了每秒非正常网络连接的个数作为控制图的统计对象，对于系统环境的安全状态还需要进一步确定。文献[13]利用 agent 技术来确定系统可能存在的环境状态，通过关联规则算法和时序序列算法从大量数据挖据若干模式，但是没有考虑到正常/异常库的划分。

3. 自适应策略

3.1. 自适应策略的必要性

目前许多入侵检测系统都是采用一种检测策略，用于检测所有环境，然而在检测强度与范围上没有改变。这使得系统的误报率与漏报率有时比较高，从而降低了入侵检测的可信度。这个问题可以通过自适应策略的思想来改善，检测策略随着环境的改变而调整，不同的环境状态对应不同的检测策略。

3.2. 自适应策略的描述

自适应策略分为状态空间以及策略空间。状态空间是用来描述系统环境，策略空间用来表示可能采用的策略。状态空间的某一环境状态与策略空间的策略存在映射关系。策略

流程图如图 1。

3.3. 状态空间与策略空间

环境状态 E 可以表示为 $E=(e_1,e_2,\cdots,e_n)$。

其中表示某 $e_i\ (1\le i\le n)$ 个环境变量，环境状态由多个环境变量组成。典型的环境变量有 cpu 的使用率，网络连接数等。

状态空间是由若干的环境状态的集合，状态空间 C 可以表示为 $C=(c_1,c_2,\cdots,c_n)$。

策略空间是由各种策略组成的集合，策略空间 P 可表示为 $P=(p_1,p_2,...,p_n)$，其中 $p_i\ (1\le i\le n)$ 表示某个具体的策略。如本文的孤立森林检测方法和谱聚类检测方法。

3.4. 自适应策略选择

安全阈值是用户和系统行为根据某种属性计数，如一种特定类型的网络连接数、企图访问文件数、访问文件或目录次数和访问网络系统次数等。对于安全阈值的判断是在一个特定的时间间隔内进行的。这个时间间隔在时间上可以是固定的，例如，阈值在每天的特定时间重置为 0；也可以在一个滑动窗口上运行，例如，每隔 1 小时进行度量。

传统的 IDS 是让所有的网络环境状态对应于一个策略，不具备自适应性。自适应策略是在构建状态空间和策略空间的基础上，对于环境变量设置阈值。大于安全阈值的情况，采用孤立森林的方法检测；在小于等于安全阈值的情况下，采用谱聚类的方法进行检测。

4. 孤立森林的检测方法

南大周志华老师在 2010 年提出一个异常检测算法 Isolation Forest，在工业界很实用，算法效果好，时间效率高，能有效处理高维数据和海量数据。算法起源于 08 年的一篇论文《Isolation Forest》[14]，这

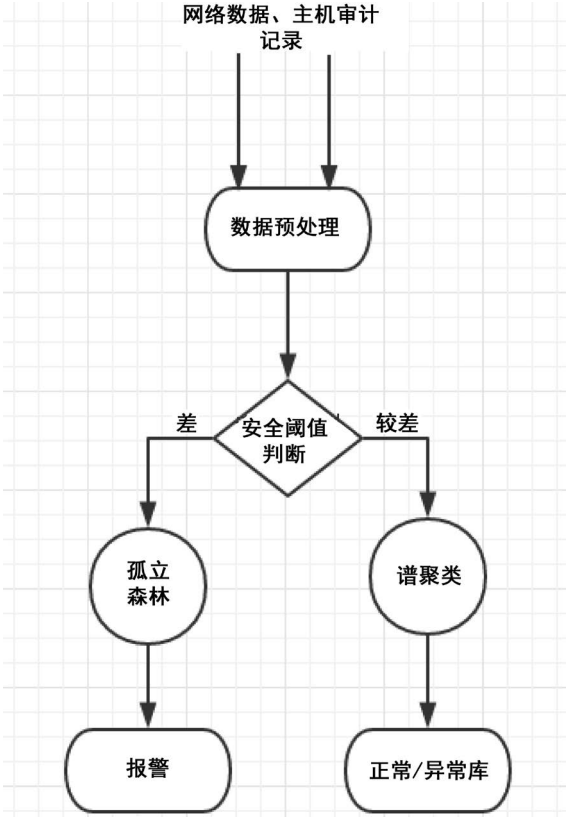


Figure 1. Strategy flow chart
图 1. 策略流程图

论文由澳大利亚莫纳什大学的两位教授 Fei Tony Liu, Kai Ming Ting 和南京大学的周志华教授共同完成，而这三人在 2011 年又发表了《Isolation-based Anomaly Detection》[15]，这两篇论文算是确定了这个算法的基础。孤立森林是由许多孤立树组成，最后综合所有孤立树的结果。

4.1. 孤立森林用于入侵检测

具有线性时间复杂度，对内存要求较小，能够处理极大的数据量和大量不相关属性的高维问题。这与入侵检测实时性、能够处理大量数据的要求相吻合，所以可以应用于入侵检测当中。

4.2. 孤立森林原理

异常数据相对于正常数据是少而且不同，所以可以使用称为孤立树(iTree)的二叉树结构来有效分割实例。由于易于分离，异常数据更有可能与 iTree 的根源接近，而正常数据更有可能在 iTree 较深端被隔离。iForest 由多个 iTree 组成，异常数据就是那些在孤立森林中具有短的平均路径。

4.3. 详细说明

4.3.1. 异常森林构建

算法步骤如下图 2 [15]：

1) 从训练数据中随机选择 ψ 个点样本点作为子样本，放入树的根节点。

2) 随机指定一个属性(attribute)，在当前节点数据中随机产生一个切割点 p——切割点产生于当前节点数据中指定维度的最大值和最小值之间。

Algorithm 1 iTree(*X*, *e*, *h*)

Input: *X*-input data; *e*-current height; *h*-height limit.
Output:an *i*Tree.

1: if e ≥ *h* OR $|X| \le 1$ then
2: return exNode $\{Size \leftarrow |X|\}$
3: else
4: Randomly select an attribute *q*;
5: Randomly select a split point p between *min* and *max* values of attribute *q* in *X*;
6: $X_l \leftarrow filter(X, q < p), X_r \leftarrow filter(X, q \ge p)$;
7: return $inNode\{Left \leftarrow iTree(X_l, e+1, h), Right \leftarrow iTree(X_r, e+1, h), SplitAttr \leftarrow q, SplitValue \leftarrow p\}$;
8: end if

Figure 2. Isolation forest algorithm
图 2. 孤立森林算法图

3) 以此切割点生成了一个超平面，然后将当前节点数据空间划分为 2 个子空间：把指定属性里小于 p 的数据放在当前节点的左孩子，把大于等于 p 的数据放在当前节点的右孩子。

4) 在孩子节点中递归步骤 2 和 3，不断构造新的孩子节点，直到孩子节点中只有一个数据(无法再继续切割)或孩子节点已到达限定高度(height limit)。

IF 采用二叉树去对数据进行切分，数据点在二叉树中所处的深度反应了该条数据的“疏离”程度，整个算法总体可以分为训练和预测两步。

孤立树训练阶段：

对于每个孤立树，从总数据中无放回抽取 n 个数据进行训练，之后随即选取一个特征，在这个特征值所有范围内随机选择一个值，对样本进行划分。小于这个值划分到节点的左边，大于等于该值的划分到节点右边，然后再对左右两个数据集重复上面的过程，直到满足终止条件一数据不可分(只有一个数据样本或全部样本相同)或达到树的高度 $\log 2^n$。

用同样的方法，随机选择不同的样本训练得到多个树。

预测阶段：

把测试数据在每个 iTree 上沿着对应的条件分支往下走，直到达到叶子节点，并记录这过程中经过的路径长度 h(x)，即从根节点，穿过中间的节点，最后达到叶子节点所走过的边的数量，最后计算出异常分数。

4.3.2. 异常分数计算

先沿着一棵 iTree，从根节点开始按不同特征的取值从上往下，直到到达某叶子节点。假设 iTree 的训练样本中同样落在 x 所在叶子节点的样本数为 T.size，则数据 x 在这棵 iTree 上的路径长度 h(x)，公式(1)：$h(x) = e + C(t.size)$。

公式中，e 表示数据 x 从 iTree 的根节点到叶节点过程中经过的边的数目，C(T.size)可以认为是一个修正值，它表示在一棵用 T.size 条样本数据构建的二叉树的平均路径长度。一般的，C(n)的计算公式(2)如下：$C(n) = 2H(n-1) - \frac{2(n-1)}{n}$。

$H(n-1)$ 可用 ln(n − 1) + 0.5772156649 估算，这里的常数是欧拉常数。数据 x 最终的异常分值[15] Score(x)综合了多棵孤立树的结果公式(3)：$Score(x) = 2^{-\frac{E(h(x))}{C(\psi)}}$。

公式中，E(h(x))表示数据 x 在多棵 iTree 的路径长度的均值，ψ 表示单棵 iTree 的训练样本的样本数，

$C(\psi)$表示用ψ条数据构建的二叉树的平均路径长度，它在这里主要用来做归一化。

从异常分值的公式看，如果数据 x 在多棵 iTree 中的平均路径长度越短，得分越接近 1，表明数据 x 越异常；如果数据 x 在多棵 iTree 中的平均路径长度越长，得分越接近 0，表示数据 x 越正常；如果数据 x 在多棵 iTree 中的平均路径长度接近整体均值，则打分会在 0.5 附近。

5. 谱聚类的检测方法

5.1. 谱聚类用于入侵检测的可行性

数据挖掘是从海量的数据中挖掘出潜在知识的过程，聚类方法是一种无监督的学习技术，可以在无标记的数据集上建立入侵检测模型，非常适应于实际情况。谱聚类算法具有可伸缩性好、能够处理不同形状的数据集、能够处理高维数据等特点，非常适合应用于入侵检测。

5.2. 谱聚类算法

谱聚类[16]是把聚类问题转化为对图的划分，把数据看成空间的点，将点之间的相似度量化为相应顶点连接边的权值，最后得到一个基于相似度量的加权图。之后采用基于图论的划分方法使得划分后的子图内部权值和尽量大，不同子图之间权重和尽量小，从而实现聚类。

由于图的划分属于一个 NP 问题，解决方法是将其转换为求解相似矩阵得到的拉普拉斯矩阵的谱分解。

第一步：根据高斯函数来得到数据的相似矩阵。

第二步：求解拉普拉斯矩阵 L。

第三步：求出矩阵 L 前 k 个特征值与之对应的特征向量。

第四步：将这些特征向量按照行标准化，形成特征矩阵 F。

第五步：将矩阵 F 的行向量转化为单位向量，得到矩阵 Y。

第六步：将 Y 每一行看作是空间的一个点，用其他聚类算法(如 K-means 算法)，得到 K 个聚类

K-means 算法步骤：首先从 n 个数据对象任意选择 k 个对象作为初始聚类中心；而对于所剩下其它对象，则根据它们与这些聚类中心的相似度(距离)，分别将它们分配给与其最相似的(聚类中心所代表的)聚类；然后再计算每个所获新聚类的聚类中心(该聚类中所有对象的均值)；不断重复这一过程直到标准测度函数开始收敛为止。一般都采用均方差作为标准测度函数。k 个聚类具有以下特点：各聚类本身尽可能的紧凑，而各聚类之间尽可能的分开。

5.3. 正常/异常库划分

因为正常行为记录数目远远大于异常行为数目，正常行为记录的聚类应该比较大分布密集，因此将聚类中数目大于 N 的聚类为正常聚类，其他的为异常，经过一定形式变换，形成正常行为库和异常行为库。

6. 数据预处理

6.1. 标准化正交处理

对原始数据进行标准化处理。经过处理的数据符合标准正态分布，即均值为 0，标准差为 1，转化函数式(4)为：

$$X = \frac{X - \mu}{\sigma}$$

其中 μ 为所有样本数据的均值， σ 为所有样本数据的标准差。这样可以去除多组数据之间的差异性，从而使得多组数据可以进行比较。

6.2. 离散数据连续化

两种属性类型：normal 设为 0，error 为 1。

3 种类型如 protocol_type:tcp,udp,icmp.

Tcp: protocol_type1 = 1, protocol_type2 = 0, protocol_type3 = 0;

Udp: protocol_type1 = 0, protocol_type2 = 1, protocol_type3 = 0

7. 实验仿真

实验环境在 PC 机上(内存为 4G，CPU 为 CORE-i53470，操作系统为 Windows 7)，采用 python 编程。

7.1. 系统状态与安全阈值

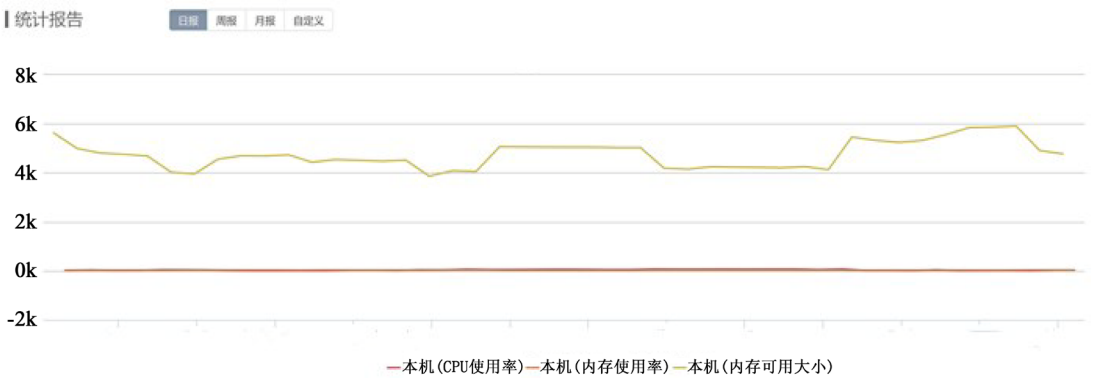
系统状态描述如图 3 所示。

安全阈值设定：可以把 30 分钟内的网络连接数设为 20 次[17]。

7.2. Isolation forest 检测

tcpdump 抓包如图 4 所示。

使用 wireshark 查看 tcpdump 抓到的包，可以看到针对同一个主域名，包含了大量的子域名请求，而



颜色	设备名称	监测点名称	监测指标	平均值	最大值	最小值
	本机	AgentWinCPU	CPU使用率	35.9228	75.000	0.760
	本机	虚拟内存	内存使用率	27.8336	35.100	21.280
	本机	物理内存	内存可用大小	4753.16	5907.000	3860.000

Figure 3. System state description
图 3. 系统状态描述图

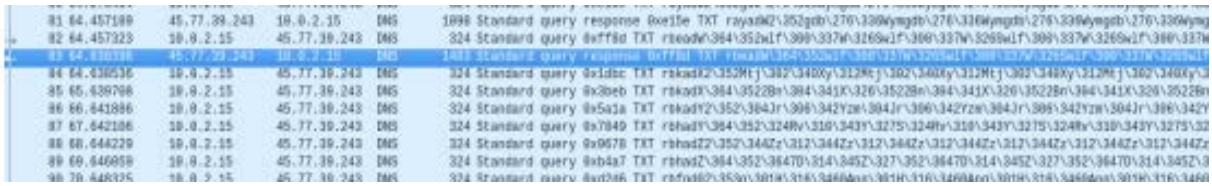


Figure 4. tcpdump packet capture
图 4. tcpdump 抓包图

这些子域名负责携带外发数据。然而 pcap 这种特殊的文件格式并不能直接作为 iforest 检测算法的输入数据，所以需将其转化为 iforest 算法所能够识别的文本文件。对其进行特征处理，从中提取解析的 dns 域名，srcip 个数，dstip 个数，报文长度等特征。

数据输入如图 5 所示。

数据输出如图 6 所示。

根据 iforest 算法中的异常分数为 1 得到可能为异常 dns 查询信息(dns 流量中可能潜在攻击，比如 dns tunnel)，如图 7 所示。

根据 wireshark 对“vxcache.com”抓包分析，可以看出一段 Label 的字节数为 0 × 80——即 128 字节。而 DNS 请求的数据结构中队 Label 的长度[18]是有限制的：Labels must be 63 characters or less (参考 RFC882 [Page 30])。Label 被允许的最大长度只有 63 字节——即 0 × 3f，只要大于这个值，即为异常。

7.3. 谱聚类

在选取的 KDD 数据集中有 43 个属性，但是有些属性对于入侵检测是无关的，Srinivas Mukkamala [19]

```
(u'abixsc.com', (24, 1, 4, 11, 0.4583333333333333, 48.0, 56.041666666666664, 1.0, 8.0, 0.0, 3.0, 0.5, 0.0, 0.20833333333333334, 0.08333333333333333,
0.004340277777777778))
(u'dqdkws.cn', (71, 2, 7, 50, 0.704225352112676, 45.0, 79.859154929577471, 1.0, 6.0, 0.0, 2.4132632507067329, 0.5915492957746479, 0.0, 0.0, 0.0,
0.0015649452269170579))
(u'tcdnvod.com', (701, 51, 17, 40, 0.05706134094151213, 55.266761768901567, 56.370898716119832, 3.1749999999999998, 17.399999999999999, 0.125,
3.4810606143066232, 0.9714693295292439, 0.0, 0.39514978601997147, 0.0442225392296719, 0.00012905890248309329))
(u'0937jyg.com', (68, 4, 7, 19, 0.27941176470588236, 46.25, 67.529411764705884, 1.0, 5.3684210526315788, 0.0, 2.2469056830015672, 0.6323529411764706,
0.0, 0.0, 0.0, 0.0015898251119236884))
(u'jcloud-cdn.com', (61, 3, 3, 11, 0.18032786885245902, 67.278688524590166, 66.311475409836063, 4.5454545454545459, 24.363636363636363,
0.18181818181818182, 3.5244668708659161, 0.4262295081967213, 0.0, 0.08196721311475409, 0.03278688524590164, 0.0012183235867446393))
(u'cmacloud.com', (545, 8, 20, 29, 0.053211009174311193, 46.315596330275227, 30.722935779816513, 1.9655172413793103, 17.793103448275861, 0.0,
3.3836270422458083, 1.0, 0.0, 0.10825688073394496, 0.022018348623853212, 0.0001980825608134618))
(u'serverss.top', (144, 1, 15, 22, 0.1527777777777778, 46.604166666666664, 50.145833333333336, 1.0, 4.5909090909090908, 0.0, 2.1594720075625,
0.5277777777777778, 0.0, 0.2777777777777778, 0.06944444444444445, 0.0007450454777231408))
```

Figure 5. DNS query information
图 5. DNS 查询信息图

```
bad domains: ('openvpn.', [81.0, 5.0, 3.0, 14.0, 0.1728395061728395, 27.493827160493826, 32.76543209876543, 3.2857142857142856, 18.214285714285715,
3.0714285714285716, 3.255427209766844, 0.0493827160493827, 0.0, 0.3950617283950617, 0.12345679012345678, 0.002245717287831163])
bad domains: ('mobily.com.sa', [16.0, 1.0, 4.0, 12.0, 0.75, 47.3125, 108.8125, 1.0, 5.333333333333333, 0.0, 1.9166666666666667, 0.6875, 0.0, 0.375,
0.375, 0.006605019815059445])
bad domains: ('vc12728.com', [40.0, 2.0, 10.0, 27.0, 0.675, 67.125, 462.85, 3.3333333333333335, 28.555555555555557, 3.3703703703703702,
3.111111111111111, 0.025, 0.0, 0.0, 0.0, 0.001862197392923651])
bad domains: ('vkcache.com', [598.0, 1.0, 2.0, 528.0, 0.882943143812709, 47.0, 161.65886287625418, 1.0, 6.0, 0.005681818181818182, 2.453875312427234,
0.22909698996655517, 0.0, 0.11371237458193979, 0.0033444816053511705, 0.00017789795773144525])
bad domains: ('nsconcreteblock.info', [18.0, 2.0, 4.0, 18.0, 1.0, 87.0, 43.5, 1.0, 37.0, 5.0, 3.823329582775343, 1.0, 0.0, 0.0, 0.0,
0.0031928480204342275])
bad domains: ('topcdn.org', [52.0, 2.0, 4.0, 13.0, 0.25, 80.92307692307692, 56.38461538461539, 1.0, 40.92307692307692, 0.0, 4.176988788169356, 0.5,
0.0, 0.28846153846153844, 0.21153846153846154, 0.001188212927756654])
bad domains: ('bilibiligame.net', [6472.0, 165.0, 17.0, 32.0, 0.004944375772558714, 46.542954264524106, 88.28522867737948, 1.0, 18.65625, 2.84375,
3.4818361348887463, 0.9610630407911002, 0.0, 0.2376390605686032, 0.0004635352286773795, 1.659883277007961e-05])
```

Figure 6. Abnormal DNS query information
图 6. 异常 DNS 查询信息图

```
0000  00 50 56 ed 16 dc 00 0c  29 67 e5 b9 08 00 45 00   .PV..... )g....E.
0010  00 b5 b6 ab 00 00 80 11  8a 51 c0 a8 01 82 c0 7e   ........ .Q.....~
0020  76 92 08 40 00 35 00 a1  22 c0 e6 1c 01 00 00 01   v..@.5.. ".......
0030  00 00 00 00 00 00 03 77  77 77 80 42 42 37 36 46   .......w ww.BB76F
0040  31 43 42 43 32 44 33 46  30 31 31 30 34 41 30 35   1CBC2D3F 01104A05
0050  30 42 33 37 38 34 39 45  44 30 36 33 44 38 44 39   0B37849E D063D8D9
0060  43 31 34 44 45 38 33 30  45 30 33 32 35 41 43 39   C14DE830 E0325AC9
0070  37 37 44 42 33 31 41 34  45 33 30 36 37 45 43 42   77DB31A4 E3067ECB
0080  34 36 36 41 46 34 46 42  31 41 35 37 44 31 41 35   466AF4FB 1A57D1A5
0090  43 38 31 43 35 36 46 45  36 45 35 33 44 42 37 42   C81C56FE 6E53DB7B
00a0  39 42 33 45 34 31 36 46  43 31 42 35 46 45 33 31   9B3E416F C1B5FE31
00b0  31 46 39 33 41 35 33 42  44 36 30 02 63 6e 00 00   1F93A53B D60.cn..
00c0  01 00 01                                          ...
```

Figure 7. Analysis of abnormal DNS query information
图 7. 异常 DNS 查询信息分析图

等人利用 svm 方法通过实验指出在 KDDcup1999 数据集中有 duration，src_bytes，dst_bytes，urgent，count，srv_count，same_srv_rate，dst_host_count，dst_host_srv_count，dst_host_same_srv_rate，dst_host_same_src_port_rate，protocol_ type 和 service，其中既包含了数值型属性，也包含了符号型属性。本文的实验也是基于这 13 个特征，采用 MATLAB 7 编程。

7.3.1. 谱聚类效果

原始数据如图 8 所示。

聚类之后如图 9 所示。

由图可以看出正常行为数量远大于异常行为数量。

7.3.2. 谱聚类算法与 k-means 算法比较

Kddcup99 数据集包括四种攻击数据集和正常数据集如表 1 所示。

KDD CUP1999 数据集是当前入侵检测领域权威的测试数据集。数据是来源于 1998 年 DARPA 入侵检测评估计划，并包含约 4,900,000 个数据实例。 对于每个 TCP/IP 连接，有 41 个特征。一些特征是基本的(如：持续时间，协议类型等)，而其他特征是通过使用一些领域知识获得的(例如：失败的登录尝试次数等)。在所有 41 个特征中，有 8 个离散特征和 33 个连续特征。攻击分为四大类：

DOS (拒绝服务)，如死亡 ping 攻击；U2R (User to Root)，如弹出攻击；R2U (远程用户)，如访客攻击;PROBING，例如端口扫描攻击。

选择 7996 个实例作为来自原始数据的训练集。在这个训练集有 7249 个正常实例和 747 次攻击事件。选择了四组数据，表 2 为测试数据集的数据量。

检测率和误报率，这是评估入侵检测指标的标准。在实验中，检测率被定义为系统检测到的入侵实例的数量除以在测试集中入侵实例数量；在测试集中。误报率定义为总数错误分类为入侵的正常实例的数量除以正常情况的总数，算法比较如表 3 所示。

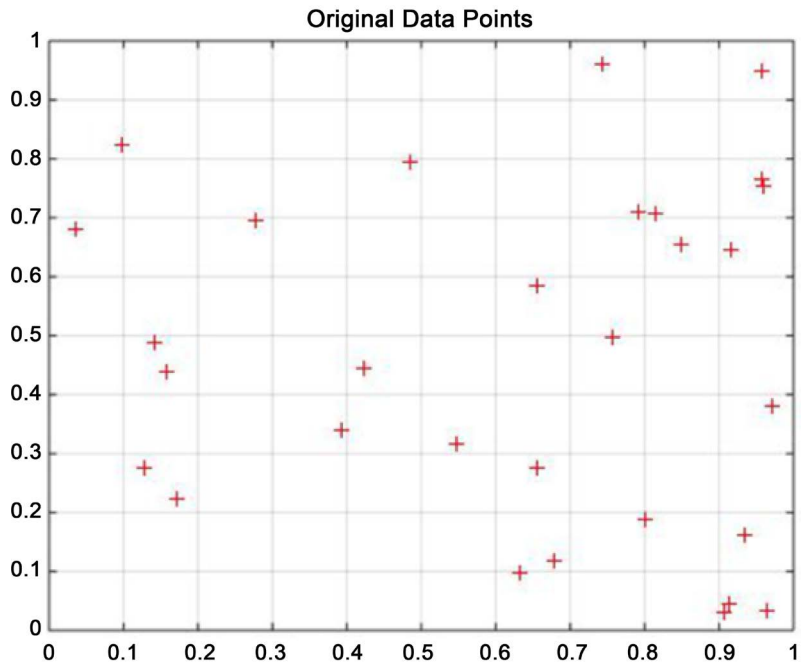


Figure 8. Raw data
图 8. 原始数据效果图

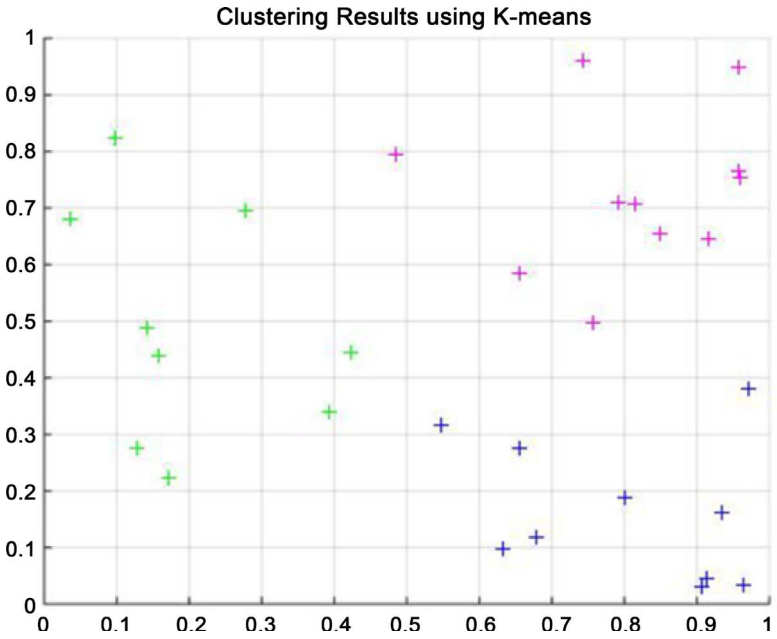


Figure 9. Clustering
图 9. 聚类后效果图

Table 1. Attack Types and Number in Data Sets
表 1. 数据集中攻击类型和数目

类别	攻击名称和数目
DOS	284. *neptune* (141), *smurf* (143).
U2R	68. *buffer_overflow* (22), *loadmodule* (2), *perl* (2), *ps* (16), *rootkit* (13), *xterm* (13).
R2U	131.*ftp_write* (3), *guess_passwd* (31), *imap* (1), *multihop* (18), *named* (17), *sendmail* (17), *phf* (2), *waremaster* (29), *xlock* (9), *snoop* (4).
PROBING	123. *ipsweep* (30), *nmap* (19), *portsweep* (32).*satan* (42).

Table 2. The amount of data in the test data set
表 2. 测试数据集中数据量

测试集	正常数据数量	异常数据数量
1	1795	204
2	1810	189
3	1845	154
4	1799	200

Table 3. Comparison of algorithm performance
表 3. 算法表现比较

算法	1 检测率	1 误报率	2 检测率	2 误报率	3 检测率	3 误报率	4 检测率	4 误报率
k-means	62.70%	0.72%	49.31%	1.10%	51.25%	1.57%	53.13%	0.95%
谱聚类	79.20%	0.67%	73.22%	0.98%	73.34%	0.93%	71.27%	1.19%

从表 3 可以看出，在检测率方面谱聚类算法优于 k-means 算法。因为 k-means 算法适合处理容易分类的数据，而那些难以分类的数据可以通过谱聚类算法进行处理，所以在入侵检测当中谱聚类具有更好的性能。

8. 总结与未来工作

本文针对传统入侵检测适应性差等问题，通过对系统环境描述来选择检测策略，将孤立森林算法与谱聚类算法相结合应用于入侵检测，最后实验表明提高了入侵检测效果。但是对系统行为记录及其分布作进一步的研究，使不同类型的行为记录能更好区分，划分出的正常/异常库能够更加准确反映系统状态。

参考文献

[1] Zuech, R., Khoshgoftaar, T.M. and Wald, R. (2015) Intrusion Detection and Big Heterogeneous Data: A Survey. *Journal of Big Data*.

[2] Zhou, C.V., Leckie, C. and Karunasera, S. (2010) A Survey of Coordinated Attacks Detection. *Computer & Security*, **29**, No. 1. https://doi.org/10.1016/j.cose.2009.06.008

[3] Julisch, K. (2003) Clustering Intrusion Detection Alarms to Support Root Cause Analysis. *ACM Transactions on Information and System Security*, **6**, 443-471. https://doi.org/10.1145/950191.950192

[4] Cuppens, F. (2001) Managing Alerts in a Multi-Intrusion Detection Environment. *Proceedings of the* 17*th Annual Computer Security Applications Conference*, IEEE, 2002, 22.

[5] Ning, P., Cui, Y. and Reeves, D.S. (2002) Constructing Attack Scenarios through Correlation of Intrusion Alerts. *Proceedings of the* 9*th ACM Conference on Computer and Communications Security*, ACM, 245-254.

[6] Ning, P., Cui, Y., Reeves, D.S. and Xu, D. (2004) Techniques and Tools for Analyzing intrusion Alerts. *ACM*, **7**, 274-318.

[7] Roshan, S. and Miche, Y. (2017) Adaptive and Online Network Intrusion Detection System Using Clustering and Extreme Learning. *Journal of the Franklin Institute*, **355**, 1752-1779.

[8] Pham, L.H., Albanese, M. and Venkatesan, S. (2016) A Quantitative Risk Assessment Framework for Adaptive Intrusion Detection in the Cloud. *The* 2*nd IEEE Workshop on Security and Privacy in the Cloud* (*SPC* 2016), 489-497.

[9] Abu Afza, A.J.M. and Uddin, M.S. (2014) Intrusion Detection Learning Algorithm through Network Mining. 16*th International Conference on Computer and Information Technology*, Khulna, 8-10 March 2014, 490-495.

[10] 杜强, 孙敏. 基于改进聚类分析算法的入侵检测系统研究[J]. 计算机工程与应用, 2011, 47(11): 106-108.

[11] Zhang, Q.-H., Fu, Y.-Z. and Xu B.-G. (2008) A New Model of Self-Adaptive Network Intrusion Detection. *IEEE Congress on Evolutionary Computation* (*CEC* 2008), 436-439.

[12] 程勇军. 基于数据挖掘的自适应入侵检测模型研究[D]: [硕士学位论文]. 重庆: 西南师范大学, 2004.

[13] Al-Yaseen, W.L., Othman, Z.A. and Nazri, M.Z.A. (2017) Real-Time Multi-Agent System for an Adaptive Intrusion Detection System. *Pattern Recognition Letters*, **85**, 56-64. https://doi.org/10.1016/j.patrec.2016.11.018

[14] Liu, F.T., Kai, M.T. and Zhou, Z.-H. (2012) Isolation-Based Anomaly Detection. *ACM Transaction on Knowledge Discovery from Data* (*TKDD*), **6**, 3.

[15] Liu, F.T., Kai, M.T. and Zhou, Z.-H. (2008) Isolation Forest. *Eighth IEEE International Conference on Data Mining*, 413-422.

[16] Zhou, Z.P., Xuan, J. and Zhao, X.X. (2017) A New Constraint Spectral Clustering Algorithm. *Control and Decision Conference* (*CCDC*), 6664-6668.

[17] 薛静锋, 祝烈煌. 入侵检测技术[M]. 北京: 人民邮电出版社, 2016.

[18] Internet Society (ISOC). (1983) Request for Comments (RFC) 882. ISI, November, p. 30.

[19] Mukkamala, S. and Janoski, G. (2002) Intrusion Detection Using Neural Networks and Support Vector Machines. *Proceedings of the* 2002 *International Joint Conference on IJCNN*'02, Honolulu, HI, 12-17 May 2002. https://doi.org/10.1109/IJCNN.2002.1007774

知网检索的两种方式：

1. 打开知网页面 http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD
   下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询
2. 打开知网首页 http://cnki.net/
   左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：http://www.hanspub.org/Submission.aspx
期刊邮箱：csa@hanspub.org