Journal of System Simulation

Volume 30 | Issue 2 Article 42

1-2-2019

Information Density based Bayes Algorithm for Cloud Platform Intrusion Detection

Du Ye
School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;

Tiantian Zhang
School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;

Meihong Li
School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;

Information Density based Bayes Algorithm for Cloud Platform Intrusion Detection

Abstract

Abstract: For getting better data classification results of cloud platform intrusion detection, and improving the detection accuracy and performance, a Bayes algorithm based on information density was proposed. *The complete probability of data characteristics was constructed, and the uncertainty of information was represented by information entropy. The information density was defined to describe the distribution of information uncertainty.* The improved algorithm was introduced, and the convergence and time complexity were analyzed. The simulation experiment results show that *the method can effectively reduce the data loss and exposethe relationship between data characteristics and data type, which can further classify the detection data of cloud platform accurately with high detection rate and low false positive rate.*

Keywords

cloud platform, intrusion detection, data characteristics, information density, Bayes algorithm

第 30 卷第 2 期 系统仿真学报© Vol. 30 No. 2
2018 年 2 月 Journal of System Simulation Feb., 2018

基于信息密度贝叶斯算法的云平台入侵检测

杜晔，张田甜，黎妹红
(北京交通大学计算机与信息技术学院，北京 100044)

摘要：为更好实现对云平台入侵检测数据的分类处理，提升检测精度和性能，提出了一种基于信息密度的贝叶斯算法。构造完整的数据特征概率集合，通过引入信息熵来表示信息的不确定度，并定义了信息密度以描述信息不确定度分布状态。对算法的收敛性和时间复杂度进行了分析，并进行仿真实验，与已有技术相比，方法可有效减少数据信息损失和描述数据特征与数据类型的概率关系，能够准确将云平台入侵检测数据分类，具有较高的检测率和较低的误报率。

关键词：云平台；入侵检测；数据特征；信息密度；贝叶斯算法
中图分类号：TP309.2 文献标识码：A 文章编号：1004-731X (2018) 02-0714-09
DOI: 10.16182/j.issn1004731x.joss.201802042

Information Density based Bayes Algorithm for Cloud Platform Intrusion Detection

Du Ye, Zhang Tiantian, Li Meihong
(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: For getting better data classification results of cloud platform intrusion detection, and improving the detection accuracy and performance, a Bayes algorithm based on information density was proposed. *The complete probability of data characteristics was constructed, and the uncertainty of information was represented by information entropy. The information density was defined to describe the distribution of information uncertainty.* The improved algorithm was introduced, and the convergence and time complexity were analyzed. The simulation experiment results show that *the method can effectively reduce the data loss and exposethe relationship between data characteristics and data type, which can further classify the detection data of cloud platform accurately with high detection rate and low false positive rate.*

Keywords: cloud platform; intrusion detection; data characteristics; information density; Bayes algorithm

引言

随着信息技术的发展，云计算已经影响了人们的生活和工作方式。一方面它为人们的生活带来无穷便利，但另一方面，云平台安全还存在着诸多待解决的问题。特别是针对云数据平台的网络攻击层出不穷，攻击手段方法不断增加，已呈智能化与协同性发展。近年来发生的亚马逊宕机事件[1]，Google Docs 用户数据泄露事件都给云计算的安全敲响警钟。云数据平台网络攻击事件的频繁发生，对广大网民网络利益产生影响，也对社会经济发展、国家信息安全造成威胁与挑战。

为了保护云平台系统资源，作为一种主动的安全防护技术，云平台入侵检测可以及时地发现各种入侵攻击以及攻击企图，并做出有效的反响应，其研究显得尤为重要。由于云平台入侵检测需要对大

收稿日期：2015-12-23 修回日期：2016-03-21；
基金项目：中央高校基本科研业务费(2014JBM030)，北京高校青年英才计划基金(YETP0548)；
作者简介：杜晔(1978-)，男，黑龙江，博士，副教授，博导，研究方向为网络安全、可靠性分析；张田甜(1990-)，男，河南，硕士生，研究方向为云平台入侵检。

量的网络访问请求数据进行分析和处理，例如对连续时间、协议类型、传送的字节数等参数进行分析，且通常处理数据是算法是复杂的，这无疑会增加检测系统分析的复杂度、时效性。因此，需要通过优化算法以适当降低问题规模，去除不相关和冗余的数据，减小问题复杂度，从而增加入侵检测的准确度。

本文提出了一种基于信息密度优化的贝叶斯算法，并应用于云平台入侵检测领域。采用信息密度对数据特征的不确定性进行密度划分，通过优化的贝叶斯算法实现模型学习，以提高模型训练和检测精度。文章的结构如下，第 1 节简单介绍了相关工作；第 2 节提出了基于信息密度的贝叶斯算法，并对算法进行了详细描述；第 3 节给出实验结果与分析；第 4 节对全文的工作进行了总结。

1 相关工作

Mazzariello C 等[2]提出了一个在云环境中基于网络的入侵检测系统，并定义了一系列的入侵规则，用来判定访问行为是否属于入侵行为，该系统虽然具有较高的检测率，但是所有入侵规则都要预定义，对于未知攻击检测的误检率较高，且该系统不能检测加密的和虚拟化的网络。Chirag N.Modil 等[3]提出了将贝叶斯分类器和基于 Snort 的网络入侵检测系统应用到云环境中，系统利用 Snort 进行云环境的网络数据采集，通过贝叶斯分类器对数据进行处理，虽然系统降低了检测的误检率和计算成本，但文中贝叶斯算法的应用要求有比较严格的独立性假设。Pardeep Kumar 等[4]提出了基于聚类方法的隐马尔科夫模型的云入侵检测技术，借助聚类技术减少了检测数据集规模，用正常的系统调用序列建立系统正常程序行为的模型，通过观察序列在正常模型下发生的概率，根据概率的大小来判断是否属于入侵行为，但 HMM 模型学习训练耗时较长，复杂度较高。Vieira[5]提出基于神经网络的云入侵检测系统，采用基于签名的神经网络模型进行行为分析，但模型参数较多，对系统的检测性能影响较大。袁凯[6]介绍了一种云计算环境下的监控系统的设计与实现，使用开源软件 Nagios 和 Ganglia 构建了 MyCloud 云平台的整体架构，对云平台的性能和故障进行监控，同时监控云平台的热点事件，并将获得的数据持久化存储，但是监控粒度和监控能力尚有不足。

2 基于信息密度的贝叶斯算法

2.1 贝叶斯算法简介

贝叶斯算法[7]将事件的先验概率与后验概率巧妙地联系起来，利用先验信息和样本数据信息确定事件的后验概率。设每个数据样本用一个 n 维特征向量来描述 n 个属性的值，即：$X=\{x_1,x_2,\cdots x_n\}$，假定有 m 个类，分别用 $C_1,C_2\cdots C_m$ 表示。给定一个未知的数据样本 X，若朴素贝叶斯分类法将未知的样本 X 分配给类 C_i，则一定是

$$P(C_i \mid X) > P(C_j \mid X)\ i, j \in [1,m], i \neq j \qquad (1)$$

根据贝叶斯定理由于 $P(X)$ 对于所有类为常数，最大化后验概率 $P(C_i \mid X)$ 可转化为最大化先验概率 $P(X \mid C_i)P(C_i)$。如果训练数据集有许多属性和元组，计算 $P(X \mid C_i)$ 的开销可能非常大，为此，通常假设各属性的取值互相独立，这样先验概率 $P(x_1 \mid C_i)$，$P(x_2 \mid C_i),\cdots$，$P(x_n \mid C_i)$ 可以从训练数据集求得。

根据此方法，对一个未知类别的样本 X，可以先分别计算出 X 属于每一个类别 C_i 的概率 $P(X \mid C_i)P(C_i)$，然后选择其中概率最大的类别作为其类别。朴素贝叶斯算法成立的前提是各属性之间互相独立。当数据集满足这种独立性假设时,分类的准确度较高，否则可能较低。另外，该算法没有分类规则输出。

2.2 基于信息密度的贝叶斯算法

贝叶斯算法对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，其概率最大的类别，就认为是此待分类项归属类别。贝叶斯算法假设属性之间相互独立，但这个假设在实际数据中往往是不成立的。此外，贝叶斯算法没有考虑数据

DOI: 10.16182/j.issn1004731x.joss.201802042

第 30 卷第 2 期 系统仿真学报 Vol. 30 No. 2
2018 年 2 月 Journal of System Simulation Feb., 2018

特征的内部联系和数据特征值的不确定性，因而忽视了数据特征值的分布关系。

以上问题使得贝叶斯算法的适用性减小。于是本文提出了基于信息密度的贝叶斯算法。其中，我们将信息密度定义为信息的分布密度即信息的不确定度。

设每个数据样本用一个 n 维特征向量来描述 n 个数据特征值，即：$X=\{x_1,x_2,\cdots x_n\}$，假定有 m 个数据类型，分别用 $C_1,C_2,\cdots,C_m$ 表示。给定一个未知的数据样本 X，计算出最大化后验概率 $P(C_i|X)$，并构造出数据样本中不同数据特征出现的概率集合如下：

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & \cdots & x_n \\ P(C_1|x_1) & P(C_1|x_2) & P(C_1|x_3) & P(C_1|x_4) & \cdots & P(C_1|x_n) \\ P(C_2|x_1) & P(C_2|x_2) & P(C_2|x_3) & P(C_2|x_4) & \cdots & P(C_2|x_n) \\ P(C_3|x_1) & P(C_3|x_2) & P(C_3|x_3) & P(C_3|x_4) & \cdots & P(C_3|x_n) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ P(C_m|x_1) & P(C_m|x_2) & P(C_m|x_3) & P(C_m|x_4) & \cdots & P(C_m|x_n) \end{bmatrix} \tag{2}$$

根据得到的数据特征概率，可求得数据的信息熵[8]，公式如下所示：

$$H(x_i)=-\sum_{i=1}^{n}P(C_i|x_i)\log P(C_i|x_i) \tag{3}$$

信息密度体现了数据特征的不确定性的分布状态，基于以上结果，可得信息密度计算公式如下所示：

$$M(x_i)=\exp(-H(x_i)) \tag{4}$$

由于不同类型的数据特征值分布存在不同的规律，某些类型的数据特征存在一些相似的分布，而信息密度是从数据特征值中求解得到。为了减小相似的数据特征对信息密度值的影响，本文引入信息密度修正值，用梯度下降法[9]求解出最合适的信息密度修正值，因为密度修正值是最优信息密度值与已得到的信息密度值的误差，而且密度修正值是一个非常小的增量，所以把信息密度值与密度修正值之和为更新后的信息密度值。根据计算出的分类结果，按照实际类型，来判断分类是否正确。若正确或当前信息密度值已经为 1，则密度修正值为 0；若错误，则密度修正值为随机值(不高于当前信息密度值的 10%)。根据新的信息密度值，计算分类结果。使用梯度下降法对密度值进行计算，求出最优的信息密度值。根据计算出的分类结果，按照实际类型，来判断分类是否正确。重复进行密度值修正计算，直到密度修正值均为 0 或者目标函数 $F(R)$ 的达到稳定状态或者达到迭代次数。最终得到的信息密度值为该数据特征的概率权重值。将测试数据记录的每个数据特征值在数据特征值概率取值表格进行匹配，得到所有数据特征的概率取值。将得到所有数据特征的概率取值与对应数据特征的信息密度值相乘再求和，得到当前测试数据记录的分类结果，算法结束，如图 1 所示。

目标函数：

$$F(R)=\frac{1}{n^2}\sum_{i,j=1}^{n}(R_i-R_j)^2 \tag{5}$$

信息密度值优化公式：

$$M_{i+1}=M_i+D_i \tag{6}$$

式中：M_{i+1} 为第 $i+1$ 次信息密度值矩阵；M_i 为第 i 次的信息密度矩阵；D_i 为第 i 次密度修正值。

$$\begin{bmatrix} m_{i,1} \\ m_{i,2} \\ m_{i,3} \\ \vdots \\ m_{i,41} \end{bmatrix}+\begin{bmatrix} d_{i,1} \\ d_{i,2} \\ d_{i,3} \\ \vdots \\ d_{i,41} \end{bmatrix}\rightarrow\begin{bmatrix} m_{i+1,1} \\ m_{i+1,2} \\ m_{i+1,3} \\ \vdots \\ m_{i+1,41} \end{bmatrix} \tag{7}$$

将数据带入各类型数据模型中，与每一个类型的数据模型进行匹配。得到测试数据的每个特征值在各类型的数据模型中出现的概率值。在将这些概率值与信息密度值相乘，得到测试数据经过数据模型[10]处理后的结果。

$$\begin{cases} R(C_1)=\prod_{i=1}^{n}P(C_1\mid X_i)M_i \\ R(C_2)=\prod_{i=1}^{n}P(C_2\mid X_i)M_i \\ R(C_3)=\prod_{i=1}^{n}P(C_3\mid X_i)M_i \\ \vdots \\ R(C_m)=\prod_{i=1}^{n}P(C_m\mid X_i)M_i \end{cases} \tag{8}$$

式中：$R(C_m)$ 表示结果；C_m 表示数据类型；$P(C_m\mid X_i)$ 表示特征 X_i 在 C_m 类型的数据模型中出现的概率值；M_i 表示特征 X_i 在 C_m 类型的数据模型中信息密度值。最终测试数据会得到多个结果 $R(C_1)$ $R(C_2)$ … $R(C_m)$，比较多个结果并取最大的 $R(C_m)$。最大的 $R(C_m)$ 所对应的 C_m 类型即为测试数据的分类结果。

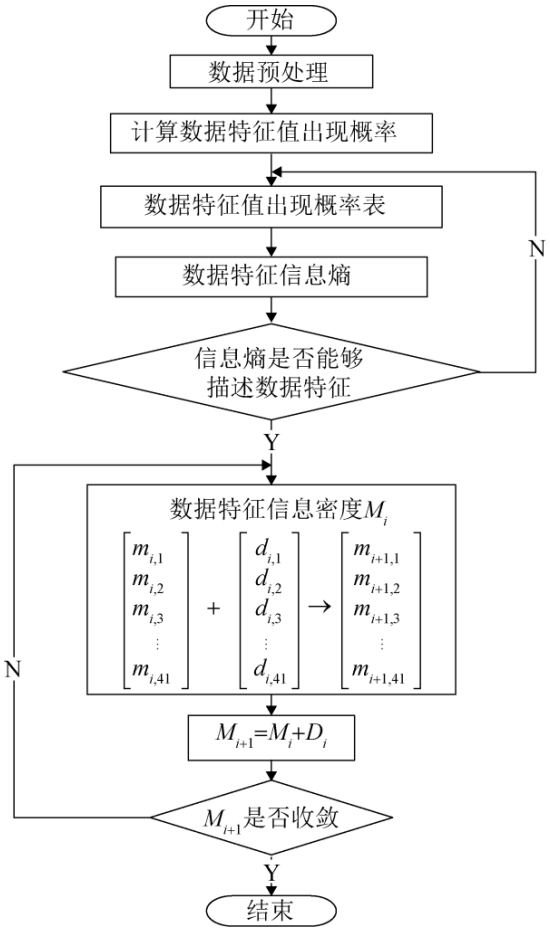


图 1 基于信息密度的贝叶斯算法流程图

Fig. 1 Flow Diagram of Information Density based Bayes Algorithm

2.3 时间复杂度分析

假设 n 代表数据集中数据的条数，m 代表每条数据中数据特征的个数。算法将数据矩阵按类别分块，不同类别的数据被分开，得到类别数据矩阵，时间复杂度为 $O(nm)$。

类别数据矩阵中有不同的数据特征，对所有数据特征计算出数据特征值概率取值表格，时间复杂度为 $O(nm)$。根据特征值的概率，求解出所有数据特征的信息熵，时间复杂度为 $O(m)$。

对所有数据特征的信息熵进行计算，求出数据特征的信息密度值，时间复杂度为 $O(m)$。

引入信息密度修正值，用梯度下降法求解出最合适的信息密度修正值，时间复杂度为 $O(nm)$。

信息密度值与密度修正值之和为更新后的信息密度值。最终得到的信息密度值为该数据特征的概率权重值，时间复杂度为 $O(m)$。

将某条测试数据记录的每个数据特征值在数据特征值概率取值表格进行匹配，得到所有数据特征的概率取值，时间复杂度为 $O(n^2m)$。

将得到所有数据特征的概率取值与对应数据特征的信息密度值相乘再求和，得到当前测试数据记录的分类结果，算法结束 $O(n^2m)$。

综上可得，算法的时间复杂度为 $3O(nm)+3O(m)+O(n^2m)$。

3 实验与分析

3.1 数据预处理

采用 KDD CUP99 数据集作为学习和检测数据来源，攻击被分为 4 类：Dos 类、Probing 类、R2L 类和 U2L 类。训练数据中包含有 22 种攻击，而测试数据中包含有 17 种未见攻击。测试环境配置包括主机采用 i7-4790 CPU，8 GB 内存，Windows7 OS，利用 Matlab 进行数据仿真处理。

对收集的数据首先进行预处理。当某条数据特征值出现空白记录，则用两个相邻数据的平均值进行填充，如果当前特征值有明显的变化趋势，则使

第 30 卷第 2 期 系统仿真学报 Vol. 30 No. 2
2018 年 2 月 Journal of System Simulation Feb., 2018

用回归拟合的方法拟合符合规律的数值，并进行填充。对数据进行观察，按照不同的类型将数据划分成数据矩阵，并计算出每种类型的数据数量。由于原始 KDDCUP 数据集[11]中的数据类型众多，为了使实验结果具有代表性，将四大类型的攻击与数据集中具体的攻击对应起来。

为了方便算法处理，对数据特征进行编码，将不具有数值意义的数据和具有数值意义的数据统一编码，使无数值意义的字符项变成具有数值意义，方便算法的处理。

3.2 数据特征的概率值

数据特征包含着许多关于数据项的信息。数据类型的不同，则数据特征也表现出不同。对已经分类的数据矩阵进行概率统计，计算不同的数据特征的数值分布结果，如表 1~3 所示。以 DOS 类型数据为例，DOS 类型数据的第 1 列的数据全为 0，共 829 次，数值 0 出现的概率为 1；第 27 列的数值类型共 6 种，出现次数分别为 749, 34, 26, 12, 6, 2 共 829 次；出现概率为 0.903 5,0.041 0,0.031 4,0.014 5,0.007 2,0.002 4 概率之和为 1。

表 1 数据类型和攻击类型表
Tab. 1 Data Type and Attack Type Table

类型	攻击类型	数量
DOS	teardrop	829
Probe	nmap	816
R2L	spy	813
U2R	perl	52
NORMAL	normal	476 1
共计		727 1

表 2 数据字符编码表
Tab. 2 Character Encoding Table

协议类型	编号	连接状态	编码	类型	编码	服务类型	编号
tcp	1	OTH	201	DOS	401	auth	101
udp	2	REJ	202	Probe	402	bgp	102
icmp	3	RSTO	203	R2L	403	courier	103
		RSTOS0	204	U2R	404	csnet_ns	104
		RSTR	205	NORMAL	405	ctf	105
		S0	206			daytime	106
		S1	207			discard	107
		S2	208			domain	108
		S3	209			domain_u	109
		SF	210			echo	110
		SH	211			eco_i	111
						ecr_i	112

表 3 DOS 类型数据特征值概率统计表
Tab. 3 DOS Data Characteristic Value Probability Statistics Table

数值类型	出现次数	出现概率
0	749	0.9035
0.01	34	0.0410
0.02	26	0.0314
0.03	12	0.0145
0.04	6	0.0072
0.05	2	0.0024
共计	829	

3.3 数据特征的信息熵和信息密度

以 DOS 类型数据为例，第 1 列的数据全为 0，数值 0 出现的概率为 1，信息熵为 0，信息熵之和为 0；则第 1 个数据特征的信息熵为 0。第 27 列的数值类型共 6 种，出现次数分别为 749,34,26,12,6,2；出现概率为 0.903 5, 0.041 0, 0.0314, 0.014 5, 0.007 2, 0.002 4；对应的信息熵为 0.132 3, 0.186 0, 0.156 7, 0.088 4, 0.005 15, 0.021 0，信息熵之和为 0.638 8；则

第 27 个数据特征的信息熵为 0.638 8，如表 4 所示。由公式（4）可得信息密度表，如表 5 所示。

表 4 DOS 类型第 27 列数据特征值信息熵统计表
Tab. 4 The 27th Column Data Characteristic value entropy statistics table of DOS

数值类型	出现概率	信息熵	第 27 特征的信息熵
0	0.9035	0.132 3	
0.01	0.0410	0.189 0	
0.02	0.0314	0.156 7	0.638 8
0.03	0.0145	0.088 4	
0.04	0.0072	0.051 5	
0.05	0.0024	0.021 0	

表 5 DOS 类型第 27 列数据特征值信息密度统计表
Tab. 5 The 27th Column Data Characteristic value Information Density Statistics Table of DOS

数值类型	出现概率	信息熵	第 27 特征的信息密度
0	0.903 5	0.132 3	
0.01	0.041 0	0.189 0	
0.02	0.031 4	0.156 7	0.527 9
0.03	0.014 5	0.088 4	
0.04	0.007 2	0.051 5	
0.05	0.002 4	0.021 0	

直接求解出来的信息密度值可能存在误差，通过对信息熵进行负指数变换可以得到信息密度值，但是数据特征值具有不确定性，可能导致信息熵存在细小的误差，进而使得信息密度值也有误差。引入了密度修正值，可以使信息密度值更为准确，具有客观性。使用梯度下降策略对信息密度值进行优化，找到最优化的密度修正值的集合 D 。

3.4 测试数据验证数据模型

使用测试数据的数据特征，与每一个类型的数据模型进行匹配。得到测试数据的每个特征值在各类型的数据模型中出现的概率值。在将这些概率值与信息密度值相乘，得到测试数据经过数据模型处理后的结果。

表 6 全体数据特征值信息密度统计表
Tab. 6 Data Characteristic value Information Density Statistics Table

特征	401 信息密度	402 信息密度	403 信息密度	404 信息密度	405 信息密度
1	1	0.895 6	0.111 1	0.019 4	0.073 4
2	1	0.471 7	1	1	0.630 2
3	1	0.332	0.385 4	0.347 2	0.627 6
4	1	0.250 6	0.946 8	0.790 4	0.926 3
5	1	0.429 6	0.018	0.029 3	0
6	0.975 8	0.846 1	0.198 2	0.009 8	0.003 6
7	1	1	1	1	1
8	0.932 1	1	1	1	1
9	1	1	1	1	1
10	1	0.986 5	0.249 5	0.155 2	0.997 1
11	1	1	1	1	1
12	1	0.965 6	1	1	0.619 1
13	1	1	1	0.130 5	0.994 7
14	1	1	1	0.369 5	1
15	1	1	1	1	0.997 1
16	1	1	1	0.626 1	0.980 4
17	1	1	1	0.273 7	0.902 6
18	1	1	1	1	1
19	1	1	1	1	0.914 1
20	1	1	1	1	1
21	1	1	1	1	1
22	1	1	0.405 4	1	1
23	0.000 8	0.011 2	0.412 6	0.235 1	0.554 1
24	0.001 3	0.231 1	0.388 8	0.409 4	0.184 5
25	0.198 3	0.009 9	0.962 2	0.676 2	0.954 2
26	1	0.495 9	0.949 2	1	0.950 5
27	0.527 9	0.009 5	0.962 2	0.626 1	0.969 7
28	1	0.376 8	0.975 4	0.790 4	0.953 8
29	0.133 3	0.097 7	0.926 3	0.536 9	0.732 2
30	0.223 4	0.154 3	0.926 3	0.497 2	0.736 3
31	1	1	0.857 7	1	0.334 6
32	0.206 7	0.751 6	0.009 7	0.075	0.000 7
33	0.000 8	0.199 3	0.001 2	0.041	0.000 5
34	0.003 2	0.215	0.031 6	1	0.002 1
35	0.017 9	0.061 1	0.147 8	1	0.021 6
36	0.003 2	0.159 8	0.174 4	0.135 3	0.089
37	1	0.986 5	0.025 5	0.303 3	0.165 8
38	0.179 4	0.008 8	0.465 9	1	0.605 7
39	1	0.489 4	0.457 5	1	0.783 6
40	0.027 2	0.003 4	0.401 2	0.315 8	0.476 3
41	1	0.376 8	0.884 9	0.315 8	0.549 2

https://dc-china-simulation.researchcommons.org/journal/vol30/iss2/42
DOI: 10.16182/j.issn1004731x.joss.201802042

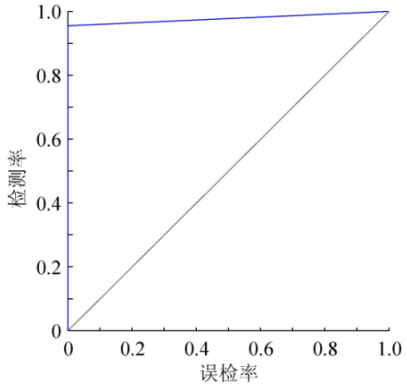
第 30 卷第 2 期 系统仿真学报 Vol. 30 No. 2
2018 年 2 月 Journal of System Simulation Feb., 2018

表 7 全体数据分类结果统计表
Tab. 7 Data Classification Statistics Table

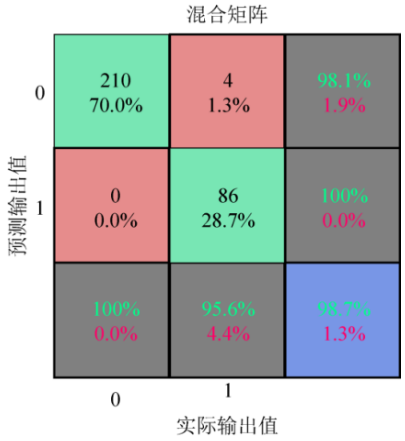
数据类型	个数	正确分类个数	准确率/%
DOS 类型	89	88	98.8
Probe 类型	90	86	95.6
R2L 类型	80	80	100
U2L 类型	4	4	100
NORMAL 类型	37	34	91.9
总数	300	292	97.3

3.5 实验结果

使用 ROC 曲线[12]反映数据模型分类器的检测率和误检率，揭示检测率和误检率的相互关系，通过将连续变量设定出多个不同的临界值，从而计算出一系列检测率和误检率，再以检测率为纵坐标、误检率为横坐标绘制成曲线；分类结果表对实验分类结果的检测率和误检率进行描述。部分实验结果如图 2~3 所示。

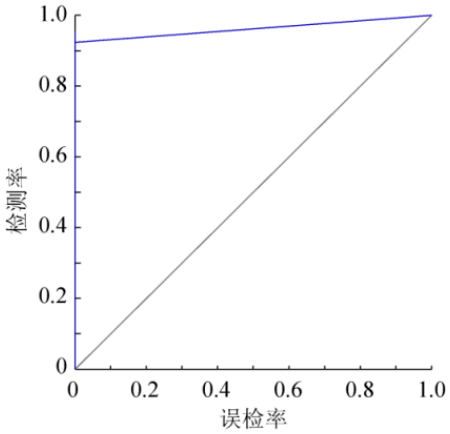


(a) Probe 数据类型的分类结果 ROC 曲线

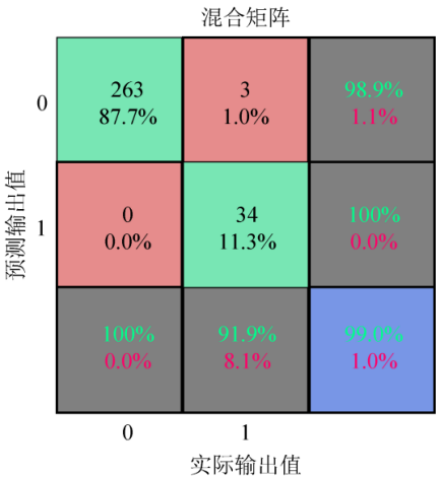


(b) Probe 数据类型的分类结果表

图 2 Probe 数据类型分类结果
Fig. 2 Classification Results of Probe Data



(a) NORMAL 数据类型的分类结果 ROC 曲线



(b) NORMAL 数据类型的分类结果表

图 3 NORMAL 数据类型的分类结果
Fig. 3 Classification Results of NORMAL Data

根据数据模型的 ROC 曲线的特点，曲线下的面积大，分析出数据模型分类的准确度较高。在 ROC 曲线上，最靠近坐标图左上方的点为检测率和误检率均较高的临界值。数据模型能够准确的检测不同的数据类型，检测率、误检率、分类准确率等评价指标都达到较好水平。Probe 类数据的分类准确率为 98.7%，Probe 类数据被误判为非 Probe 类数据的比率为 1.3%，非 Probe 类数据被误判为 Probe 类数据的比率为 0%；NORMAL 类数据的分类准确率为 99%，NORMAL 类数据被误判为非 NORMAL 类数据的比率为 1.0%，非 NORMAL 类数据被误判为 NORMAL 类数据的比率为 0%。

针对样本集中其它类型数据也分别作了测试，其中 DOS 类数据的分类准确率为 99.7%，DOS 类

数据被误判为非 DOS 类数据的比率为 0.3%，非 DOS 类数据被误判为 DOS 类数据的比率为 0%；R2L 类数据的分类准确率为 100%，R2L 类数据被误判为非 R2L 类数据的比率为 0%，非 R2L 类数据被误判为 R2L 类数据的比率为 0%；U2L 类数据的分类准确率为 100%，U2L 类数据被误判为非 U2L 类数据的比率为 0%，非 U2L 类数据被误判为 U2L 类数据的比率为 0%；

用本文提出基于信息密度的贝叶斯算法处理数据，得出分类准确度为 97.33%，用时 37 s；用神经网络算法[13]处理数据，得出分类准确度为 97.00%，用时 50 s；用 K-邻近算法[14]处理数据，得出分类准确度为 94.93%，用时 32 s；将 3 种算法的分类准确率进行对比，结果如表 8 所示。

表 8 算法分类准确率对比表
Tab. 8 Comparison Table of algorithm classification accuracy rate

分类算法	分类准确度/%	算法用时/s
基于信息密度的贝叶斯算法	97.33	37
神经网络算法	97.00	50
K-邻近算法	94.93	32

由表 8 可知，基于信息密度的贝叶斯算法对数据的分类准确度较高，能够较准确的实现对数据分类。

4 结论

云平台入侵检测样本数据规模的大小，影响到算法的性能和检测率。为了更好地对样本数据进行分类，提出了一种基于信息密度的贝叶斯算法，采用信息熵描述数据特征的不确定性，构造出完整的数据特征概率集合，通过引入信息熵来表示信息的不确定度，并定义了信息密度以描述信息不确定度分布状态。实验结果表明，该方法可有效减少数据信息损失，能够准确将云平台入侵检测数据分类，具有较高的检测率和较低的误报率。实验表明，与改进前及其他相关算法相比，本文算法可有效减小信息损失和描述数据特征与数据类型的概率关系，准确地实现云平台入侵检测数据的分类。

参考文献:

[1] 娄伟. 云计算环境下用户信息安全策略研究[D]. 长春: 东北师范大学, 2013.
Lou Wei. Research on User Information Security Strategy in Cloud Computing Environment[D]. Changchun: Northeast Normal University, 2013.

[2] Esposito M, Oliviero F, Mazzariello C, et al. Evaluating Pattern Recognition Techniques in Intrusion Detection Systems[C]//Proceedings of the 5th International Workshop on Pattern Recognition in Information Systems, USA, 2005: 144-153.

[3] Modi C N, Patel D R, Patel A, et al. Bayesian Classifier and Snort based network intrusion detection system in cloud computing[C]//. Proceedings of 2012 International Conference on Computing Communication & Networking Technologies, India, 2012: 1-7.

[4] Kumar P, Nitin N, Sehgal V, et al. A novel approach for security in Cloud Computing using Hidden Markov Model and clustering[C]//. Proceedings of 2011 World Congress onInformation and Communication Technologies, India, 2011: 810-815.

[5] Vieira K, Schulter A, Westphall C, et al. Intrusion Detection for Grid and Cloud Computing[J]. It Professional (S1520-9202), 2010, 12(4): 38-43.

[6] 袁凯. 云计算环境下的监控系统设计与实现[D]. 武汉: 华中科技大学, 2012.
Yuan Kai. Design and Implementation of Monitoring System Based on Cloud[D]. Wuhan: Huazhong University of Science and Technology, 2012.

[7] 范仕伦, 薛天俊, 夏玮. 基于贝叶斯算法和费舍尔算法的垃圾邮件过滤系统设计与实现[J]. 信息网络安全, 2012, 24(9): 18-22.
Fan Shilun, Xun Tianjun, Xia Wei. Spam Email Filter System based on Bayesian Algorithm and Fisher Algorithm Design and Implementation[J]. Netinfo Security, 2012, 24(9): 18-22.

[8] 赵姝, 吕靖, 张燕平. 不完整数据集的信息熵集成分类算法[J]. 模式识别与人工智能, 2014, 25(3): 193-198.
Zhao Shu, Lv Jing, Zhang Yanping. Information Entropy Ensemble Classification Algorithm for Incomplete Data[J]. Pattern Recognition and Artificial Intelligence, 2014, 25(3): 193-198.

（下转第 730 页）