基于推进贝叶斯分类法的入侵检测引擎研究

张元清　包骏杰
（重庆教育学院计算机与现代教育技术系　重庆 400067）

摘　要　为了提高贝叶斯分类法的准确率，设计了基于推进技术的贝叶斯分类法，并将推进贝叶斯分类法应用到入侵检测引擎中，并设计了基于推进贝叶斯分类的入侵检测引擎。通过实验表明，此检测引擎可以有效的将入侵行为与非入侵行为进行分类，与传统贝叶斯分类法的检测引擎相比，此引擎对数据的分类有更高的准确率。

关键词　入侵检测，数据挖掘，贝叶斯，推进

Research on Intrusion Detection Engine Based on Boosting Bayesian Classification Algorithm

ZHANG Yuan-Qing　BAO Jun-Jie
(Department of Computer and Modern Education Technology, Chongqing Education Collage, Chongqing 400067)

Abstract　To improve the accurate of Bayesian algorithm, a new Bayesian classification algorithm which based boosting was designed. The new Bayesian algorithm had been used in Intrusion Detection System, and an engine of Intrusion Detection System based the algorithm had been designed. Experiments show that the boosting Bayesian classification algorithm is more accurate than traditional Bayesian classification algorithm based Data Mining detection.

Keywords　Intrusion detection, Data mining, Bayesian, Boosting

近年来各种网络攻击方式层出不穷，网络安全面临着严峻的挑战。误用检测对新的入侵方法检测不到，其检测入侵行为的能力取决于规则库的新旧程度。异常检测是依据任何一种入侵行为都能由其偏离正常或者所期望的系统和用户活动规律而被检测出来。如果发现了当前状态偏离了正常的模型状态，则发出警告信号，任何不符合以往活动规律的行为都将被视为入侵行为。

异常检测不依赖于规则库，特别适合当今新型攻击手段层出不穷的网络中使用。但是，异常检测其模型复杂，系统设计与调试都很不方便。为了克服异常检测的缺点，人们开始将不断发展的数据挖掘(Data Mining)中的分类技术应用于入侵检测中。1999 年，Wenke Lee 在其博士论文中给出了用数据挖掘技术建立入侵检测模型的过程[1]，如图 1 所示。

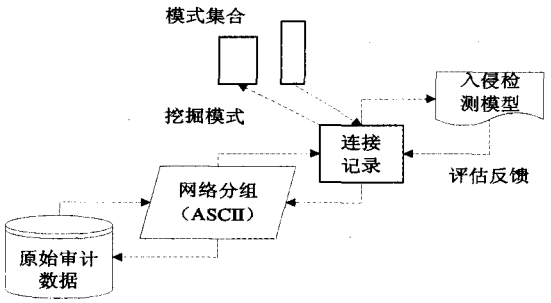

图 1　用数据挖掘建立入侵检测模型的过程

数据挖掘技术中的贝叶斯分类是基于统计学的分类法，理论上讲，与其它分类法相比，贝叶斯分类法有最小的出错率[2]。本文首先设计了入侵检测系统中贝叶斯分类算法，然后通过推进技术提高分类法的准确率。

1　贝叶斯分类算法设计

贝叶斯分类基于贝叶斯定理，本文根据入侵检测系统的特点，设计了其在入侵检测系统中的分类算法。算法主要工作过程如下：

(1)数据样本用一个 n 维特征向量 $X=\{x_1,x_2,\cdots,x_n\}$ 表示，分别描述样本的 n 个属性。

(2)假定有 m 个类 $C_1,C_2,\cdots,C_m$，给定一个未知的数据样本 X，分类法将预测 X 属于条件 X 下具有最高后验概率的类。即贝叶斯分类将未知的样本分类给 C_i，当且仅当

$$P(C_i|X)>P(C_j|X),1\leqslant j\leqslant m,j\neq i \tag{1}$$

其中 $P(C_i|X)$ 最大的类 C_i 称为最大后验假定。根据贝叶斯定义

$$P(C_i|X)=\frac{P(X|C_i)P(C_i)}{P(X)} \tag{2}$$

(3)由于 $P(X)$ 对于所有类为常数，只需要 $P(X|C_i)P(C_i)$ 最大即可。因为 $P(C_i)=s_i/s$，其中 s_i 是类 C_i 中的训练样本数，s 是总样本数，所以我们只须对 $P(X|C_i)$ 求最大。

(4)为降低 $P(X|C_i)$ 的开销，根据入侵检测中类条件独立的条件，得到

$$P(X|C_i)=\prod_{k=1}^{n}p(x_k|C_i) \tag{3}$$

概率 $P(x_1|C_i)$，$P(x_2|C_i)$，…，$P(x_n|C_i)$ 可以由训练样本估值，其中 $P(x_k|C_i)=s_{ik}/s_i$，s_{ik} 是属性 A_k 上具有值 x_k 的类 C_i 的训练样本数，而 s_i 是 C_i 中的训练样本数。

(5)对未知样本 X 分类，对每个类 C_i，计算 $P(X|C_i)P(C_i)$。样本被指派到类 C_i，当且仅当

$$P(X|C_i)P(C_i)>P(X|C_j)P(C_j)$$
$$1\leqslant j\leqslant m,j\neq 1 \tag{4}$$

张元清　讲师，研究方向：现代教育技术。

虽然在理论上，贝叶斯分类法有最小的出错率，但是，在实践中贝叶斯分类相比其它分类法并没有优势。这主要是由于对其应用的假设条件的不准确性造成的[2]。因此，应该寻求一种解决方案提高贝叶斯的分类的准确率。

2 基于推进贝叶斯分类法的入侵检测引擎设计

推进技术是数据挖掘中一个提高分类法准确率的普遍技术[2]。将推进技术用于贝叶斯分类，可以达到提高检测正确率的目的。

2.1 推进模型

推进的工作原理如图2。对给定的样本集合 S，有放回的随机选取 t 次，由每个训练样本集 S_t 在某个具体分类算法指导下学习，得到一个分类模型 C_t，并根据分类模型对样本的分类准确率更新其权值。这里，每个分类法的表决是其准确率的函数。对一个未知的样本 X 分类，每个分类模型 C_t 返回它的类预测，算作一票。最后，统计得票，将得票最高的类赋予 X。

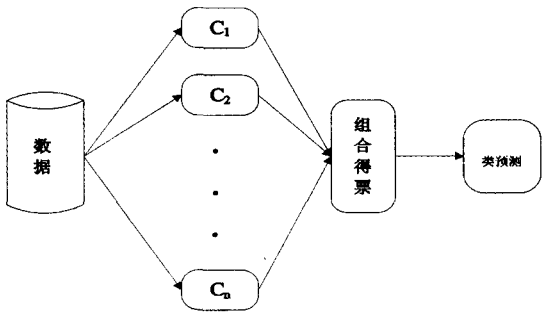

图2 推进工作原理图

2.2 推进算法设计

推进算法，其主要思想是：根据每个分类模型的正确率，分别给予它不同的权值(正确率越高，权值越大)，将分类结果进行加权求和，选择值最大的作为最终分类。其针对入侵检测系统的分类模型生成算法主要流程如下：

(1)对有 s 个样本的集合 S，重复 k 次，每次从 S 中有放回的任意抽取总样本数的 $\gamma_s(0<\gamma<1)$ 作为训练集 S_t(t 为整数且将 $\alpha_s(0<\alpha<1)$ 作为测试集 E_t(t 为整数且 $1\leqslant t\leqslant k$)。权值 w_t 初始为 1(t 为整数且 $1\leqslant t\leqslant k$)。

(2)将抽取的 k 组样本作为 k 组训练集分别对当前的分类模型进行训练，这样一共得到 k 组新分类模型 $C_i(1\leqslant i\leqslant k)$。

(3)随机从测试集选择样本 X 作正确性测试，将每个分类法 C_i 产生的类预测送到表决模块，将得票最多的类作为分类结果，将分类结果与 X 所属类相同的分类法的权值 w_i 加1。重复第3步 $n(n<k)$ 次，转4。

(4)定义阈值 $u(u<n)$，将权值小于 u 的分类法 C_i 去除，得到最终分类模型。

对于推进算法中的参数 γ 和 u，用户可以根据需要设定。如果用户希望测试集小一点，可以将 γ 变大，反之亦然；如果用户希望算法生成较多的分类模型，可以将 u 设小一点，反之亦然。

2.3 基于推进贝叶斯分类法的入侵检测引擎结构

本文将推进贝叶斯分类法引入了入侵检测引擎，构造的基于推进贝叶斯的入侵检测原型系统的基本结构如图3所示。

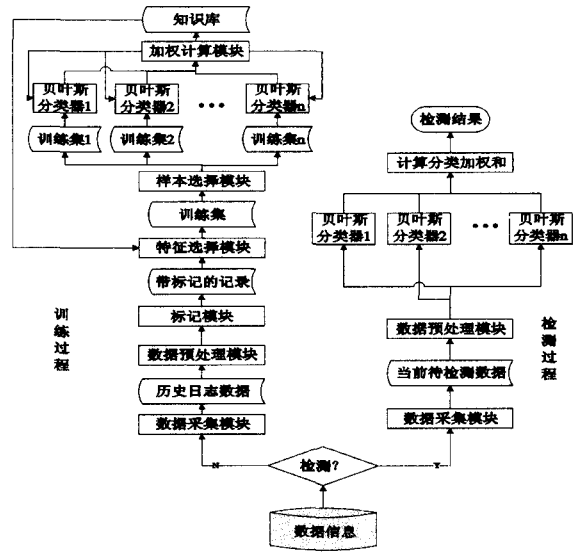

图3 基于推进贝叶斯分类法的入侵检测引擎

图3中整个系统的工作过程包括两个部分：训练过程和检测过程。在训练过程中，系统使用大量带标记的网络训练数据集，然后对数据集进行有放回的随机选取，以组成不同的数据子集，每个数据子集对对应的分类器进行训练，通过不断循环反馈使得分类器可以分辨或预测哪些行为是正常的，哪些行为是不正常的。然后，通过测试集对分类器进行测试，在测试过程中，系统利用训练过程中得到的知识库，使用训练好的分类器对测试数据进行分类，从而判断出当前行为是正常行为还是异常行为，并根据测试结果设置分类器的权值。

2.4 训练过程

在训练过程中，数据采集模块中使用脚本程序自动收集主机日志信息，它截获的数据不能直接用分类算法进行分析，因此首先需要对它进行预处理，从中提取有意义的特征。数据预处理模块负责对采集到的数据进行预处理，以便把它们转换为引擎能够识别的格式。

标记模块的作用是对训练数据进行标记，以便区分出正常记录和攻击记录。将带标记的数据输入给特征选择模块。由于日志记录可能包含很多特征，而在检测中并不需要分析全部特征，即会有冗余的特征存在，因此应该把这些冗余特征去掉。

对训练集进行随机选取，组成多个训练子集，每个训练子集对分类器进行训练，在训练过程中，需要依据训练结果指导特征选择模块进行更进一步的特征选择，不断优化所采用的特征集合，该过程循环往复，直到得到良好的分类结果为止。从而形成稳定的知识库。该知识库将用于测试过程。

随机选取样本对各个分类器进行测试，并且根据测试反馈的分类结果对各个分类器的权值进行设置。该过程进行多次，得到有统计意义的权值。

2.5 检测过程

检测过程中要使用训练过程所生成的知识库和各个分类器。检测进行时，实时采集网络流数据，并通过数据预处理模块将其转换为分类器能识别的格式，这两个部分的工作原理与前述相同。

预处理后将各个需要检测的数据包送到分类器中，由各个分类器依据相应的知识库对其进行分类检测，并将检测结果送到加权计算模块，它通过计算加权和，得到值最大的分

类，从而决定是否有违反安全策略的入侵行为发生。

3 实验分析

为了测试推进贝叶斯分类法的入侵检测引擎的检测率，我们设计了推进贝叶斯分类法的入侵检测系统（简称为BBIDS），与直接使用贝叶斯分类法的入侵检测系统作比较实验，并进行实验结果分析。

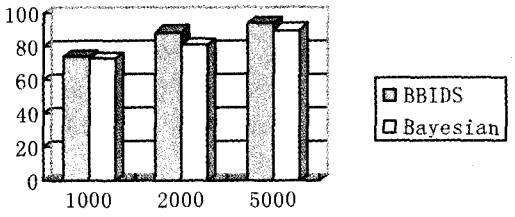

图4 比较实验结果

实验数据选用KDD Cup 1999网络数据集[3]，为了方便实验我们从中选取了10000条数据，这些数据中包含100条异常数据。从实验数据中抽取4组数据组成3个训练子集（T_1，T_2，T_3）和一个测试集（E_1），3组训练子集分别包括1000个、2000个和5000个带标记数据，测试集包含1000个不带标记的数据。实验中分别使用T_1，T_2，T_3训练基于引导聚集ID3分类法的入侵检测系统和ID3分类法的入侵检测系统，然后使用训练好的分类模型对E_1进行对比测试，得到3个测试结果。如图4所示。

通过对图4的观察，可知训练数据越丰富时，引擎检测检测率越高，BBIDS的检测率一直比贝叶斯高，当训练数据为5000时，BBIDS的检测率达到了94.07%。因此，使用推进技术可以提高分类的正确率。使用BBIDS的入侵检测系统与使用贝叶斯的入侵检测系统相比具有更高的检测率。

结论 推进贝叶斯分类算法的学习需要搜集大量的网络访问的数据进行学习，借此来训练入侵检测系统的学习模型。这是一项非常复杂的工作，但是从实验结果看来，运用加权引导聚集，通过加权学习，可以提高检测率、减小误判，因此该方法运用到入侵检测中取得了良好的效果。

参考文献

1 Lee W. A data mining framework for constructing features and models for intrusion detection systems: [dissertation of Doctor of Philosophy]. Columbia University, 1999
2 Han Jiawei, Kamber M. Data mining: concepts and techniques [M]. San Francisco: Morgan Kaufmann Publishers, 2001. 185～219
3 http://kdd.ics.uci.edu
4 唐正军. 网络入侵检测系统的设计与实现. 电子工业出版社. 2002

（上接第61页）

址利用率的衡量标准并不是通常意义的百分比利用率，而是用HD比率来衡量[6]，其定义为

$$\text{HD比率}=\frac{\text{Log}(\text{已经分配的地址空间})}{\text{Log}(\text{最大可分配地址空间})} \quad (5)$$

其中，“地址空间”表示/48的客户数量。HD比率等于0.94，相应换算成各前缀的实际使用百分率。可以看出，实际IPv6使用率一般不会高于50%，在宽松的后继申请利用率的条件下，自适应二分法按照2的幂次分解进行地址申请的机会小得多，可以预计其聚合性能将会更好。

结束语 IP地址分配方法是影响各级路由表增长速度的重要因素，地址使用单位从方便管理和控制运营成本的角度也寻求最大限度的聚类所拥有的IP地址段。然而，我国运营商在现行地址分配政策下，只能周期性地申请地址，满足网络扩容的需求，迫切需要一种有效的地址分配方法来指导其进行IP地址分配。模拟实验表明，本文所提出的改进的二分地址分配方法具有良好的聚类特性，可以有效地减少地址碎片并提高地址利用率，为不同层次的地址分配机构实际分配IPv6地址提供了有益的参考。

参考文献

1 IPv4路由表情况. http://bgp.potaroo.net/
2 Xu Z, Meng X, Zhang L, et al. Impact of IPv4 Address Allocation Practice on BGP Routing Table Growth. IEEE Computer Communications Workshop(CCW), Oct. 2003
3 Wang Mei. A Growth-based Address Allocation Scheme for IPv6 Networking. LNCS, 2005, 3462: 671～683
4 APNIC. Policies for IPv4 address space management in the Asia Pacific region. December 2005
5 Huston G. Consideration of the IPv6 Allocation Unit Size. http://www.potaroo.net/drafts/draft-huston-ip6-allocation-unit-00.txt, June 2005
6 Durand A. RFC 3194, The H-Density Ratio for Address Assignment Efficiency: An Update on the H ratio

（上接第79页）

点乘运算，而新方案的每个用户要进行N次G_1上的点乘运算和N次公钥加密运算，并且广播的数据量也扩大了N倍。但这是为原方案的DC-Net增加匿名可撤销特性所付出的代价，并且这些操作只需在初始化阶段进行一次。如果采用群签名法实现可撤销匿名性[3,4]，则需对每个匿名消息都进行群签名和签名验证，并要增加复杂的零知识证明协议，与本文的方法相比，其额外开销要大得多。

结束语 对一种新型的DC-Net匿名通信方案进行了改进，改进方案用很低的代价实现了可撤销的发送者匿名性，使得在至少t个权威执行成员的参与下可以追踪系统中任意匿名消息的发送者。可以证明，改进后的匿名性基于双线性Diffie-Hellman判定问题的困难性，在安全性上与原方案是相当的。

参考文献

1 Chaum D. The dining cryptographers problem: unconditional sender and recipient untraceability. Journal of Cryptology, 1988, 1(1): 65～75
2 Chaum D. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 1981, 24(2): 84～88
3 Stefan K, Rolf W, Hannes F. Revocable anonymity. In: Proceedings of ETRICS 2006. Heidelberg: Springer-Verlag, 2006. 206～220
4 von Ahn L, Bortz A, Hopper N. Selectively traceable anonymity. In: Proceedings of PET2006. Cambridge, UK: Springer-Verlag, 2006. 586～615
5 Golle P, Juels A. Dining cryptographers revisited. In: Advances in Cryptology: Eurocrypt' 2004. Berlin: Springer-Verlag, 2004. 456～473
6 Waidner M. Unconditional sender and recipient untraceability in spite of active attacks. In: Advances in Cryptology: Eurocrypt' 89. Berlin: Springer- Verlag, 1989. 302～319
7 Boneh D, Franklin M. Identity based encryption from the Weil Pairing. SIAM J of Computing, 2003, 32(3): 586～615
8 Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612～613
9 Cramer R, Damgaard I, Schoenmakers B. Proofs of partial knowledge and simplified design of witness hiding protocols. In: Advances in Cryptology: Crypto' 94. Berlin: Springer-Verlag, 1994. 174～187
10 Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of ACM CCS' 93. New York: ACM, 1993. 62～73