文章编号:1001 - 9081(2007)05 - 1041 - 03

基于加权多随机决策树的入侵检测模型

赵晓峰,叶 震
(合肥工业大学 计算机与信息学院,安徽 合肥 230009)
(qboy_best@163.com; yezhen1952@yahoo.com.cn)

摘 要:传统的决策树分类方法(如 ID3 和 C4.5)对于相对小的数据集是很有效的。但是,当这些算法用于入侵检测这样的非常大的数据时,其有效性就显得不足。采用了一种基于随机模型的决策树算法,在保证分类准确率的基础上,减少了对系统资源的占用,并设计了基于此算法的分布式入侵检测模型。最后通过对比试验表明该模型在对计算机入侵数据的分类上有着出色的表现。

关键词:决策树;入侵检测;分辨矩阵;随机决策树
中图分类号: TP309 **文献标识码**:A

Intrusion detection model based on weighted multi-random decision tree

ZHAO Xiao-feng, YE Zhen
(*School of Computer and Information, Hefei University of Technology, Hefei Anhui* 230009, *China*)

Abstract: The traditional decision tree category methods (such as: ID3, C4.5) are effective on small data sets. However, when these methods are applied to massive data of IDS, its effectivity will get influenced. In this paper, a random model based decision tree algorithm was applied, and an intrusion detection model based on it was provided. It is verified by experiment that this model is evidently powerful for IDS.

Key words: decision tree; intrusion detection; discernibility matrix; random decision tree

0 引言

从本质上讲,入侵检测是一个分类问题,对其研究的重点是寻找一个能够将入侵和正常网络行为准确分类的方法。传统入侵检测的评判标准着重于系统对入侵的检出能力,即怎样降低误报和漏报率。而目前随着网速的不断提高,对 IDS 系统的实时性的要求也越来越高。所以,寻求一个在占用少量系统资源(如内存和网络资源)的条件下能对大量复杂数据进行快速准确分类的算法也成为当今入侵检测研究的核心问题。

决策树分类方法,因其分类准确、稳定性强、构造简单等优点已被广泛应用于数据的分类决策中。但因为其结构特点,对解决高维的大数据量问题存在着很多不足[1]。在传统单决策树的基础上,多决策树方法得到了越来越广泛的应用[2~4]。相对于单决策树来说多决策树既提高了分类准确率也减小了树的规模,但也可以看出这种多决策树方法的基础还是传统的单决策树,并不能完全避免单决策树的弊端。

研究表明将随机的方法引入决策树的构建之中可以有效地提高决策树分类的准确率[5~8]。本文在 IBM T. J. Watson Research 的 Wei Fan 等人的研究基础上提出了一种加权多随机决策树的构造算法——WRDT (Weighted Random multi-Decision Tree) 算法,将属性权值的概念引入到多随机决策树的构建之中,并基于此算法设计出一套分布式入侵检测系统模型。最后,通过试验证明该系统具备对高维的大数据集进行快速、准确分类的能力,同时也具有相当低的系统资源占有率。

1 加权多随机决策树算法

1.1 属性权值的确定

本文采用分辨矩阵结合专家经验的方法来确定数据集中每个属性的权值。

定义 1 信息系统

一个信息系统可定义为 $L = (U, Q, V, f)$,其中 $U = \{x_1, \cdots, x_n\}$ 是论域,$Q = A \cup D$ 是属性集合,其中 A 为条件属性集,D 为决策属性集,V 为属性取值的集合,f 是 $U \times Q \rightarrow V$ 的映射。

定义 2 分辨矩阵

一个信息系统的分辨矩阵是一个 $|U| \times |U|$ 的对角矩阵。其中每一项定义为

$$c_{ij} = \begin{cases} \{a \in A \mid a(x_i) \neq a(x_j)\}, & d(x_i) \neq d(x_j), d(x) \in D \\ \varnothing, & d(x_i) = d(x_j), d(x) \in D \end{cases}$$

通过分析发现:属性在分辨矩阵中出现的次数越多属性越重要;包含属性的数据项越短属性越重要。基于以上两点给出计算属性重要度的方法。

设 $w(a_i)$ 是属性 a_i 的重要度(权值)

1) 初始时对所有 $a_i \in A$ 令 $w(a_i) = 0$。

2) 对分辨矩阵中下对角阵的每一项 c_{jk} 计算

$$w(a_i) = w(a_i) + \frac{|A|}{|c_{jk}|}, a_i \in c_{jk}, 0 < k < j \leqslant |U|$$

其中,$|A|$ 是所有属性的基数,$|c_{jk}|$ 是分辨矩阵中 c_{jk} 的基数。

对于数据量非常大的信息系统可以将整个信息系统划分为 M 个子系统,对于第 $m(m < M)$ 个子系统求出属性 a_i 的权值 $w_m(a_i)$。则 a_i 在整个系统中的权值 $w(a_i) = \dfrac{\sum_m w_m(a_i)}{M}$。

设由专家经验给出的权值为 $w'(a_i)$,则属性 a_i 的权值 $w_{a_i} = w(a_i) + w'(a_i)$。

1.2 加权多随机决策树

设 $\mathbb{F} = \{F_1, F_2, \cdots, F_K\}$ 为数据集的属性集合,$D_i = \{d_{i1},$

收稿日期:2006 - 11 - 08

作者简介:赵晓峰(1978 -),男,浙江绍兴人,硕士研究生,主要研究方向:计算机网络、信息安全; 叶震(1953 -),男,安徽无为人,副研究员,主要研究方向:信息安全.

$d_{i2},\cdots,D_{im_i}\}$ 为属性 F_i 所有不同取值的集合，r_i 为属性 F_i 的权值，表示该属性的重要度，r_i 越大属性重要度越高。$\{T_1,T_2,\cdots,T_N\}$ 为生成的 N 棵随机决策树，$Y=\{y_1,y_2,\cdots,y_n\}$ 为数据分类。U 为数据集合（即定义 1 中的论域），其中 x_i 对应于集合中的第 i 条记录。训练数据和测试数据是以 (x_i,y_j) 的形式给出的，表示第 i 条记录属于 y_j 类。树的结构描述如下：树的每一层对应于一个选中的分裂属性；树中的每个节点表示一个问题；树中的分支表示节点分裂属性 F_i 的可能取值 d_{ix}。首先，用上节的方法计算每个属性的权值。然后，建立 N 棵空决策树，每棵树对应于随机选择的 K' 个属性。每棵树的权值为该树所有属性权值的平均。在树建成后，用训练数据更新每棵树相应节点的统计数据。经过训练后的树的每个节点保存落入该节点的不同类训练数据的数量。具体过程（不包含剪枝）如算法 1 所示。

算法 1

train$(S,\mathbb{F},N,R,K')$

输入：

S 为训练数据集；

$\mathbb{F}=\{F_1,F_2,\cdots,F_K\}$ 为数据集的属性集合

N 为要创建的随机决策树的数量；

$R=\{r_1,r_2,\cdots,r_K\}$ 为上一小节中得出的属性权值的集合，其中 r_i 为属性 F_i 的权值；

K' 为每个决策树包含的属性数量。

输出：经过训练的 N 棵随机决策树 $\{T_1,T_2,\cdots,T_N\}$ 和每棵树的权值 w_i

begin

for $i\in\{1,\cdots,N\}$ do

在 $\mathbb{F}$ 中随机选择 K' 个属性，并按 D_i 中值的多少对所选属性进行递增排序得到属性集 $f_i=\{f_{i1},f_{i2},\cdots,f_{iK'}\}$；

$w_i=\dfrac{\sum_{f_i} r_i}{K'}$；

以 f_{i1} 为根节点建立一棵空决策树，树的第 j 层对应的分裂属性为 f_{ij}；

End

For *each*$(x,y)\in S$ do

For $i\in\{1,\cdots,N\}$ do

设 $n[y]$ 为节点中保存经过该节点且属于第 y 类的训练数据的数量；

从根节点出发搜寻该条测试数据所对应的叶子节点，$n[y]\leftarrow n[y]+1$；

End

End

Return$\{T_1,T_2,\cdots,T_N\}$

End

以下算法为分类算法：

算法 2

classify$(\{T_1,T_2,\cdots,T_N\},x)$

输入：

N 棵经过训练的决策树；

x 为一条待分类的数据。

输出：x 属于各类的概率 $P[y]$

begin

对于每一棵随机决策树 T_i，$P_i(y\mid x)$ 表示当前测试数据 x 属于类 y 的概率，w_i 表示该决策树的权值。$n[y]$ 是 x 所到达 T_i 的叶子节点中所保存的统计数；

$$P_i(y\mid x)=\frac{n[y]}{\sum_y n[y]}\times w_i;$$

$$P[y]=\frac{1}{N}\sum_{i=1}^{N}P_i(y\mid x);$$

对于所有的分类计算 $P[y]$，就是该测试数据属于各类的概率；

end

2 加权多随机决策树在入侵检测中的应用

从上一节的算法中我们可以看出加权多随机决策树有以下优点：

1）建树和训练用时少。传统的决策树构造方法（如：ID3，C4.5）在训练时为了选择下一节点需要计算属性信息熵，所以要对数据集进行多次扫描。而多随机决策树算法只需要对数据集进行一次扫描就可完成训练，用时比传统决策树要少很多。

2）占用内存空间少。我们对比一下未经剪枝的随机决策树和 ID3 决策树的空间使用情况。考虑最一般的情况，假设全部属性数为 K，每个属性有 d 个取值。我们为每棵随机决策树选取 $K'=\dfrac{K}{2}$ 个属性，共建立 N 棵树，则总节点数为 $n=N\times\sum_{i=0}^{K'-1}d^i$。而相应的 id3 树的总节点数为 $m=\sum_{i=0}^{K-1}d^i$。可以看出，随着 K 的增加，m 与 n 的差值是以指数增加的。并且因为随机决策树的各节点只保存落入该节点的数据的数量，所以训练数据的多少对树节点的大小没有影响。

3）因为最后的判断需要综合所有树的分类结果，某一棵树的错误判断对整体的结果不产生决定性的影响。并且，因为引入了属性权值的概念保证了重要属性对分类的贡献。

4）结果是概率的形式给出的。

5）可以将不同的树分置于不同的主机之上进行并行分类，然后将结果汇总，则总的分类时间大大减少。

6）用 WRDT 算法生成的决策树具备增量学习的特性，当有新的数据要对树进行训练时不需要对已生成好树的结构进行改变，也就是说新的训练可以在以前的训练基础上进行。

入侵检测中庞大的数据量和复杂的数据形式是很多数据挖掘算法不能有效应用于入侵检测的原因之一。而通过分析以上特性，我们发现第（1）、（2）条特性保证随机决策树能够处理大量的复杂数据，弥补了传统决策树很难处理大规模数据的不足。第（3）条特性保证了采用加权多随机决策树算法构建的入侵检测系统的稳定性。基于多 agent 的入侵检测系统是目前入侵检测系统结构的一个发展方向，而应用第（4）、（5）条特性我们可以建立起一个基于多随机决策树的多 agent 入侵检测系统。其中每个 agent 包含一棵随机决策树，且每一个 agent 只处理一棵随机决策树的分类，能快速的判断入侵。各随机决策树的分析结果是以概率的形式给出的，保证了整个系统进行全局分类的灵活性。

3 系统构架

3.1 系统中各部件说明

数据采集 agent 负责从系统中采集相关信息（包括主机和网络的信息）并将其送至相应的入侵检测 agent。

每一个入侵检测 agent 封装一棵随机决策树（由算法 1 生成）负责对相应 ICA 所采集的数据的信息进行分析以发现是否有入侵发生。多个负责相同任务的 IDA 被分为同一组，每一组 IDA 将其检测结果（被检测事件是某类入侵的概率）发送给组内的分析 agent 进行分析。多个 IDA 可以驻于同一主机中。

组内的 SA 对同组 IDA 的检测结果进行分析（运行算法 2），位于组上层 SA 对组内分析 agent 的上传结果进行进一步分析以从更高的层次对系统的安全状况进行监视。多个 SA

可以驻于同一主机中。

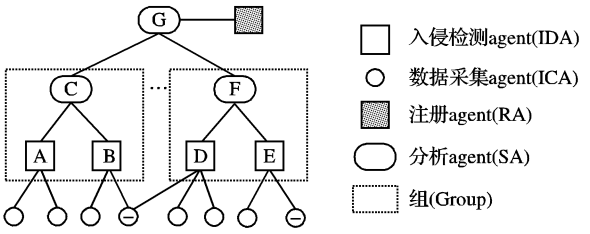

图 1　基于加权多随机决策树的入侵检测系统构架

注册 agent 保存系统中每个 agent 所需的资源及所能提供的资源。

3.2　系统检测过程

初始化时,系统中的所有 agent 将其本身的信息(包括:该 agent 的名称、所需的数据、能够提供的数据、agent 的功能及检测算法等)上传给注册 agent,当一个入侵检测 agent 需要某些数据时它就向注册 agent 查询哪些 agent 有所需的数据,然后直接与这些 agent 通讯。这样整个系统的资源得到了共享,并且注册 agent 只传送注册信息(当检测 agent 得到信息后就不再与注册 agent 通信),不会造成注册 agent 的网络负担过于沉重的问题。

每个 ICA 将其收集的系统信息上传给相应的入侵检测 agent 来进行分析,多个执行相同任务的 IDA 被分为一组,同组中 IDA 运行的随机决策树由算法 1 生成,每个组拥有一个 SA 来收集并分析(采用算法 2)同组中 IDA 的检测结果,并将其分析结果上传给上层 SA。上层 SA 对其下所有分析 agent 上传的结果进行进一步的分析以获取更大范围网络的安全情况。当系统发现入侵时采用相应的响应策略。

4　试验结果

4.1　试验数据

本文的数据采用 1998DARPA 入侵检测评估数据。该数据集是由美国 DARPA ITO 和空军研究实验室资助,由 MIT 林肯实验室负责创建的,也是世界上第一个用于入侵检测研究的标准数据集(http://www.ll.mit.edu)。数据集共提供了 7 周的训练数据和 5 周的测试数据。训练数据包括大约 5 百万条连接记录,测试数据包括大约 3 百万条连接记录。共包括四大类入侵:Probe、DoS、R2L 和 U2R,为了避免偶然性共进行 12 次分类测试,每次试验从 1998DARPA 入侵检测评估数据的训练数据集中随机抽取 200000 条记录进行训练,并且随机从测试数据集中抽取 50000 条记录进行测试。

4.2　试验结果

由于本试验的目的在于测试 WRDT 算法的分类正确率及并行分类的效率,试验只构建了图 1 中的一个组,并且在 SA 中未使用其他数据挖掘算法。本文使用 7 台 Pentium 台式机(CPU Intel 赛扬 2.4GHz,内存 DDR256MB,操作系统 Windows XP)构建图 1 中的网络结构,其中 5 台用于驻留 IDA,一台用于 SA,一台注册服务器。采用随机抽取的 50000 条测试数据作为模拟网络输入。试验每次建立 10 棵加权随机决策树,每棵决策树对应一个 IDA。试验的软件环境采用 Java 2 实现,编程环境为 Eclipse。

KDDCUP'99 所提供的试验数据是对 1998DARPA 入侵检测评估数据进行预处理后给出的。所以分类正确率的对比试验采用 KDDCUP'99 的获胜算法。其结果如表 1 所示。

模型构建及分类时间的对比试验采用 Weka 所提供的 C4.5 算法,试验缓冲区设为 128MB ~ 512MB。采用 30000 条数据进行训练时 WRDT 模型构建平均时间为 1.8s,而应用 C4.5 算法模型构建时间为 12.5s。随着训练数据的增加 WRDT 算法的模型构建时间呈线性增加:当训练数据为 200000条时 WRDT 算法的模型构建时间为 14.3s,而 C4.5 算法的构建时间为 520s;当训练数据超过 250000 条时 WRDT 算法仍可运行,而 C4.5 算法会产生内存溢出错误。KDDCUP'99 冠军的模型构建时间超过 24h(试验环境:CPU 2×300Mhz,内存 512MB)。10000 条数据的平均分类时间为 0.06s,而 C4.5 的平均分类时间为 0.23s。每棵未经剪枝的随机生成树(9 层)的平均大小为 0.37MB。

表 1　WRDT 算法与 KDDCUP'99 冠军算法正确率对比

方法	Normal(%)	DoS(%)	Probe(%)	U2R(%)	R2L(%)
WRDT	99.23	99.58	86.01	47.44	25.67
KDD	99.45	97.69	87.73	26.32	10.27

通过对比可以看出加权多随机决策树算法的分类准确率与 KDD CUP 获胜算法的结果基本持平。但是因为试验充分利用了多随机决策树的并行分类特性,其分类时间相对于传统的决策树算法有显著提高,并且因为算法所占系统资源很少,使系统具备快速处理大数据集的能力。

5　结语

本文将权值的概念引入到随机决策树的构建之中,提出了一种加权多随机决策树方法,并基于该方法提出了一套入侵检测模型。将该模型用于入侵检测的分类之中,解决了传统决策树方法难以处理入侵检测中大量复杂数据的问题。通过试验证明该模型对于判别网络入侵具有很高的分类准确率,对系统资源的占用也相对较小,并且分类结果以概率给出,提高了系统灵活性。

参考文献:

[1] HAN JW, KAMBER M. 数据挖掘: 概念与技术[M]. 范明, 孟小峰, 译. 北京: 机械工业出版社, 2001.

[2] FREUND Y. Boosting a Weak Learning Algorithm by Majority[J]. Information and Computation, 1995, 121(2): 256 - 285.

[3] FREUND Y, SCHAPIRE RE. Experiments with a New Boosting Algorithm[A]. Proceedings of the International Conference in Machine Learning[C]. San Francisco, CA, 1996. 148 - 156.

[4] BREIMAN L. Bagging Predictors[J]. Machine Learning, 1996, 24(2): 123 - 140.

[5] AMIT Y, GEMAN D. Shape quantization and recognition with randomized trees[J]. Neural Computation, 1997, 9(7): 1545 - 1588.

[6] BREIMAN L. Randomizing outputs to increase prediction accuracy [J]. Machine Learning, 2000, 40(3): 229 - 242.

[7] FAN W, WANG HX, YU PS, *et al.* Is random model better on its accuracy and efficiency[A]. Proceedings of Third IEEE International Conference on Data Mining (ICDM-2003)[C]. 2003.

[8] BREIMAN L. Random forests[J]. Machine Learning, 2001, 45(1): 5 - 32.

[9] 尹阿东. 分类发现的决策树技术研究[D]. 北京: 京科技大学, 2004.

[10] HU KY, LU YC, SHI CY. Feature ranking in rough sets[J]. AI Communications, 2003, 16(1): 41 - 50.

[11] FAN W, GREENGRASS E, MCCLOSKEY J, *et al.* Effective Estimation of Posterior Probabilities: Explaining the Accuracy of Randomized Decision Tree Approaches[A]. Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)[C]. 2005. 154 - 161.

[12] 蒋见春, 冯登国. 网络入侵检测原理与技术[M]. 北京, 国防工业出版社, 2001.

[13] http://www.ics.uci.edu/~kdd/databases/kddcup99/kddcup99.html[EB/OL], 2006 - 10 - 10.