入侵检测中的数据预处理问题研究

陈晓梅
(广东财经职业学院信息管理系　广州 510420)

摘　要　本文重点讨论入侵检测中的数据预处理问题。针对数据预处理的主要内容，给出了聚类要素的格式化处理方法，并将数据约简和规则提取结合到一起，提出了一种基于粗糙聚类方法的入侵检测预处理聚类器。最后用同一个入侵检测系统对预处理后与预处理前的检测结果进行了对比，结果表明该聚类器可有效提高入侵检测的效果。

关键词　入侵检测，数据预处理，子系统，粗糙聚类，算法

Study on the Data Preprocessing in Intrusion Detection

CHEN Xiao-Mei
(Department of Information Management, Guang dong Finance and Economics College, Guangzhou 510420)

Abstract　In this paper, the problem of data preprocessing in Intrusion Detection is mainly discussed. A network security solution including data preprocessing sub-system is proposed with explaining its tasks. The method of formatting clustering elements is also given. Then an intrusion detection preprocessing algorithm based on Rough Clustering method is presented with its implementation steps. In the end, a comparison between preprocessing and non-preprocessing is done to show that the algorithm is effective through running the same Intrusion Detection System.

Keywords　Intrusion detect, Data preprocessing, Sub-system, Rough clustering, Algorithm

1　引言

当前的入侵检测系统中大多采用异常检测方法以发现一些新的未知的入侵行为。但是，由于正常行为模型的建立完全依赖于对训练数据集中正常数据样本的学习，因此保证该数据集的洁净性，即不包含任何异常数据，对建立一个实用的入侵检测系统是至关重要的。实际上，要为系统的学习收集这样一个洁净数据集往往是不太容易的，一旦有入侵数据被作为正常数据出现在训练数据集中，必然导致该类入侵行为及其变种都将被系统视为正常数据，这种情况显然是非常危险的。因此，为了使入侵检测系统更好地发挥作用，必须对原始数据进行过滤、清理，去除可能会对系统产生不良影响的"脏"数据，这就是入侵检测中的数据预处理问题。

目前有关网络安全的研究主要集中于入侵检测和入侵防御两个方面。尽管在这两个方面的确还有很多问题尚待研究，但网络流数据的预处理问题却是关系到入侵检测效果的首要因素。事实上，一个完整的网络安全系统不仅应包括入侵检测系统(IDS)和入侵防御系统(IPS)，还应该包括前期的数据预处理系统(Data Preprocessing System, DPS)。一个带有数据预处理功能的网络安全解决方案如图1所示。互联网上的网络流数据首先进入到数据预处理子系统，经过该系统对原始数据的过滤、约简和聚类能够形成一定的规则集，作为入侵检测子系统判断某行为是否为恶意入侵行为的标准。而入侵防御子系统则利用其防火墙技术及时地中止入侵行为的发生和发展，实时地保护内部网络系统不受实质性攻击。由此可见，由数据预处理子系统、入侵检测子系统和入侵防御子系统形成了三位一体的网络防护格局，能够最大化地保证系统安全。

2　入侵检测中的数据预处理

2.1　带有数据预处理子系统的网络安全解决方案

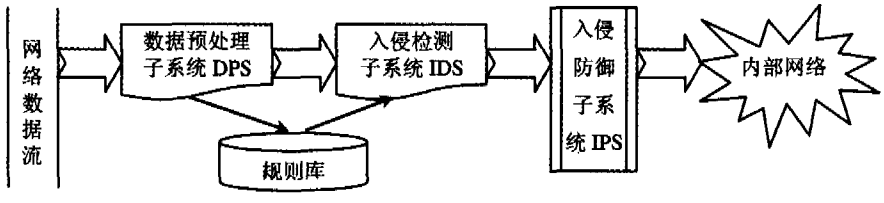

图1　带有入侵检测数据预处理子系统的网络安全解决方案

2.2　入侵检测中数据预处理的内容

在入侵检测活动中，数据预处理的对象一般分为基于主机的数据源和基于网络的数据源两种。一般说来，网络数据源的活动都会反映在主机类型的数据源(即系统日志)中，因此本文就以系统日志作为数据预处理的研究对象。具体来说，主要有以下四个步骤：数据过滤，格式化处理，数据约简和规则提取。

(1)数据过滤　入侵检测中的数据量一般都非常大，而且

陈晓梅　硕士，主要研究方向为计算机应用开发。

是动态更新的。过大的数据量可能会降低后续入侵检测的效率和效果。数据过滤的目的就是减少入侵检测系统直接处理的数据量。因为有些"脏"数据或者信息不完整的数据对入侵检测是没用的,在处理前就应该去掉,这样既可以减少存储空间又可以缩短处理时间,降低后面子系统的负荷。但数据过滤必须谨慎进行,以免删掉有用的数据。

(2)格式化处理　这一步是指对源数据进行格式化处理。基于主机的数据源格式一般都比较规范,具有固定的字段信息。而基于网络的数据源一般都是通过抓包工具(如 Wincap)获取的,数据比较零乱,缺乏统一的格式,如果不进行格式化处理将会对后续的入侵检测系统造成很大的影响。通过格式化处理,我们就可以把网上传输的原始数据包和系统日志数据转换成具有固定格式的事件序列,形成"规范"的数据库。

(3)数据约简　数据约简就是通过对数据集合的分析,在不丢失主要信息的情况下剔除无关紧要的数据,而只保留那些含有重大信息的数据。进行数据约简的原因同样是因为系统日志的数据量太大了,包含着大量的正常行为和可疑行为。而一些外来的特征使得那些可疑行为的模式更难被发现,这些难以被人们发现的特征之间又存在着复杂的关系。因此,对入侵检测系统,尤其是在需要进行实时检测的情况下,适当的数据约简是必需的。

(4)规则提取　规则提取就是在前面两步的基础上对经过初步处理的日志数据进行分析,发现具有哪些特征的属于正常行为,哪些属于可疑的入侵行为,从而产生诸如"if then"的规则集。也就是说,规则提取的目的就是得到一些结论,如果该行为具有某些特征,那么很有可能是入侵行为。当然这些结论除了需要通过训练数据进一步检验之外,更需要在实际数据集中得到真实的检验。目前的规则提取方法有两种,一种是基于训练集的,即由训练集的数据产生规则,再通过实际数据进行检验校正。这种方法得出的规则对实际数据的适应性较差,规则的动态提取性能也不强,在系统的日志数据动态增加的情况下,此种算法的实用性受到了一定的限制。第二种方法就是基于自组织聚类的方法,聚类可以用来发现数据中的隐性模式和入侵行为的显著特征。相对于基于训练集的方法,聚类可以省却训练这一步,直接对实际数据进行自组织分析,无论是从规则提取效果还是从性能上都有了很大的改进。而且聚类方法对动态增加的数据也具有较强的处理能力。因此本文主要采用聚类方法进行入侵检测数据的预处理。

3　基于粗糙聚类方法的入侵检测预处理聚类器

上述四个数据预处理内容中,数据过滤的任务比较简单,只需通过简单的过滤算法就可实现,因此本文主要着眼于数据约简和规则提取以及此前的数据格式化处理方面。

3.1　聚类要素的格式化处理

在聚类分析中,聚类要素的选择是十分重要的,它直接影响分类结果的准确性和可靠性。在入侵检测分类研究中,被聚类的对象常常由多个要素构成。不同要素的数据往往具有不同的单位和量纲,其数值的差异可能很大,这就会对分类结果产生影响。因此当分类要素的对象确定之后,在进行聚类分析之前,还要对聚类要素进行格式化处理。

假设有 m 个聚类的对象,每一个聚类对象都由 $x_1, x_2, \cdots, x_n$ 个要素构成。首先将各要素取值按照一定的离散化方法离散成可以进行代数处理的数值数据。

(1)总和标准化,分别求出各聚类要素所对应的数据总和,以各要素的数据除以该要素数据的总和,即

$$x_{ij}' = x_{ij} / \sum_{i=1}^{m} x_{ij}$$

其中　$i=1,2,\cdots,m; j=1,2,\cdots,n$,下同。

这种标准化方法所得的新数据 x_{ij}' 满足 $\sum_{i=1}^{m} x_{ij}' = 1$;

(2)标准差的标准化,即

$$x_{ij}' = \frac{x_{ij} - \bar{x_j}}{s}$$

其中,$\bar{x_j} = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$, $S_j = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_{ij} - \bar{x_j})^2}$

由这种标准化方法所得的新数据 x_{ij}',各要素的平均值为 0,标准差为 1,即有

$$\bar{x_j}' = \frac{1}{m}\sum_{i=1}^{m} x_{ij}' = 0, S_j = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_{ij}' - \bar{x_j}')^2} = 1;$$

(3)极大值标准化,即

$$x_{ij}' = \frac{x_{ij}}{\max\{x_{ij}\}}$$

经过这种标准化所得的新数据,各要素的极大值为 1,其余各数值小于 1。

(4)极差的标准化,即

$$x_{ij} = \frac{x_{ij} - \min\{x_{ij}\}}{\max_i\{x_{ij}\} - \min\{x_{ij}\}}$$

经过这种标准化所得的新数据,各要素的极大值为 1,极小值为 0,其余的数值均在 0 与 1 之间。

3.2　基于粗糙聚类方法的入侵检测预处理聚类器

对聚类要素进行格式化处理之后,源数据就具备进行后续处理的条件了。目前在有关入侵检测的研究中,对源数据的数据约简和规则提取都是分开进行的,即分别开发适用于这两个不同阶段的算法[1~3]。但这种处理方式存在着一些弊端。第一,时间复杂度很高。一个算法有时需要扫描数据库多遍,而两个阶段的不同算法需要扫描数据库的次数就更多,无形中增加了数据预处理的时间成本;第二,在规则提取的问题上,应该采取"宁多勿漏"的原则。就是说宁可提取的规则多一些,也不能把危险的行为漏过去。如果撇开后面的规则提取阶段独立进行数据约简,有可能会将对规则提取有用的信息"约简"掉,从而造成一些"危险"行为被漏掉,无疑会对系统安全造成很大影响,这显然已经偏离了入侵检测系统进行数据预处理的初衷。

针对上述问题,我们提出了一种基于粗糙聚类的入侵检测方法。该方法将粗糙集理论与聚类方法有机地结合到了一起,既考虑到了扫描数据库的时间复杂度问题,又兼顾了规则的完整性。有关粗糙集理论和聚类方法的基础知识请参考文[4],本文不作赘述。由于判断是否为恶意入侵行为是后续入侵检测子系统的工作,本算法的任务只是形成待检测的类别,故采用粗糙集中的信息表而不是决策表的方式进行聚类。该算法具体步骤如下:

输入:信息表 $T=\langle U, A\rangle$,其中 $U=\{u_i, i=1,2,\cdots,m\}$ 为元组集,$A=\{a_j, j=1,2,\cdots,n\}$ 为要素集;阈值参数:分类质量[5] γ_0,类中心最大距离 D。

输出:该信息表的聚类结果 Clu。

①预选 N_c 个初始聚类中心 $C_k (k=1,2,\cdots,N_c)$,为了尽量不漏掉危险规则,此处不预先指定聚类中心的数目,初始位置

可以从元组集中任选；

②对于要素集 $A=\{a_1,a_2,\cdots,a_n\}$，计算每一要素 a_j 的分类质量 $\gamma(a_j)$。If $\gamma(a_j)<\gamma_0$，则 $A=A-\{a_j\}$，$j=1,2,\cdots,n$；否则，转③；

③计算每一元组与初始聚类中心的距离 $d(u_i,U)$，$i=1,2,\cdots,m$；

④If $d(u_i,U)>D$，则 $i+1$，转⑤；否则，转⑥；

⑤修正各聚类中心值：$c_k=(\sum_{u\in U}u)/N_j$，$k=1,\cdots,N_c$，形成新的聚类中心 N_c'；

⑥$Clu_i=Clu_i\cup\{u_i\}$，即第 i 个聚类增加一个元组；

⑦计算全部聚类中心的距离：$D_{ij}=\|c_i-c_j\|$，$i=1,\cdots,N_c-1$；$j=i+1,\cdots,N_c$；

⑧If $D_{ij}>D$，停止；否则，转⑨；

⑨计算 Clu_i 与 Clu_j 之间的平均距离，并以此作为调整后的聚类中心；转③继续执行。则集合 $Clu=\{Clu_i,i=1,2,\cdots\}$ 即为最终的聚类结果。

本算法中采用粗糙集中关于属性分类质量作为属性重要度的定义，与后续聚类过程形成一致，并以此作为是否约简的依据。同时以类中心最大距离作为判断是否形成新类的依据，在不指定聚类数量的前提下，可以最大限度地形成完整聚类，为后续的入侵检测提供足够的信息源。分类质量的计算方法参见文[5]，类中心间的距离采用常用的两个要素之间的欧几里德距离。

4 应用实例

为了验证本算法的有效性以及产生对比效果，本文选用UCI提供的KDDCup99数据库作为实验数据集，下载地址为http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html。该数据集包括正常数据和入侵数据，共有22类入侵行为，比较典型的入侵行为有DDos、R2L、Snmpget、Probe等。每一条数据有41个要素。为了缩短实验时间，本文随机抽取了一部分数据进行处理。根据本文所述的数据预处理步骤，分别对源数据集进行数据过滤和格式化处理，并用Matlab语言实现上述算法，对处理后的数据库进行数据约简和规则的聚类提取。约简后的结果如表1所示。从表1的结果我们可以看出，处理后的数据无论是从要素方面还是从元组方面都比之前有了减少，而攻击类别的数量却没有减少，而这正是算法中没有预先指定聚类中心数目的结果。

表1 处理前后的源数据对比

	要素数量	元组数量	攻击类别数量
处理前	41	452	20
处理后	37	378	20

注：阈值 $\gamma_0=1.25$，$D=0.89$。

为了比较数据预处理前后的效果，我们采用文[6]提出的入侵检测系统对上述经过预处理的数据集进行分类，并与该文的研究结果进行对比，如表2所示。可以看出，使用相同的入侵检测系统对经过预处理的数据集进行分类，结果明显要好于未经过预处理的情况。而这正体现了入侵检测数据预处理子系统的效果。

表2 处理前后对KDDCup99数据库的分类结果对比

	检测率	误警率
处理前	67.2%	0.8%
处理后	75.2%	0.54%

需要指出的是，本算法中的两个阈值参数：分类质量 γ_0 和类中心最大距离D的选取非常重要，如果选取不当会严重影响到该算法的效果。本文是通过反复试验才确定了一个最适当的阈值。一般而言，这两个参数的取值应该根据聚类要素格式化的结果确定。

结束语 当前的很多入侵检测算法都对原始数据有着相当苛刻的要求，或者是需要完全洁净的数据，或者是需要对所有样本数据作出正确的标识，因此入侵检测中的数据预处理是不可或缺的。本文针对上述情况，提出了一种基于粗糙聚类方法的入侵检测预处理聚类器。该方法将粗糙集的属性重要度和聚类中的距离概念结合起来，将数据约简与规则提取有机地结合到了一起，大大提高了入侵检测的效率和效果。就同一入侵检测系统对预处理前数据集和预处理后数据集的实验结果表明，基于粗糙聚类的预处理聚类器明显提高了检测率，降低了误警率。尽管增加了数据预处理步骤不可避免地会带来整个系统处理时间的增加，但在网络入侵日益严重的今天，适当地以时间代价换取可靠性代价在很多情况下还是值得的。因此该方法在网络入侵检测领域具有广泛的应用前景。

但是，该方法还存在着一些亟待解决的问题，如阈值参数的确定还存在着较大的随意性。因此，下一步的工作将是研究如何对上述阈值进行合理确定的问题，同时如何与后续的入侵检测子系统更好地协同工作，以提高预处理聚类器的实用性能，也是需要进一步考虑的问题。

参考文献

1 王国胤. Rough集理论与知识获取. 西安：西安交通大学出版社，2001

2 王亚英. 基于粗集理论的知识发现方法研究：[博士学位论文]. 上海：上海交通大学，2000

3 Hu X H, Cereone N. Learning in relational databases：A rough set approach. Intl. Journal of Computational Intelligence, 1995, 11(2):323～338

4 曾黄麟. 粗集理论及其应用. 重庆：重庆大学出版社，1996

5 张祥德，张巍，刘玉蓉. 数据挖掘分类问题的贪婪粗糙集约简算法. 东北大学学报(自然科学版)，2001，22(5)：580～583

6 高飞. 数据挖掘在入侵检测特征与规则集辅助生成中的应用：[硕士学位论文]. 天津：天津大学，2004

7 牛建强，曹元大. 基于数据挖掘的IDS日志数据分析处理. 计算机应用研究，2003(9)：82～84

8 蒋建春，马恒太. 网络安全入侵检测：研究综述. 软件学报，2000，11(11)：1460～1466

9 Axelsson S. The base-rate fallacy and its implications for the difficulty of intrusion detection. In: Tsudik, Ged. Proc. of the 6th Conf. on Computer and Communication Security. New York: ACM Press, 1999. 1～7

10 Hsu C W, Lin C J. A comparison of methods for multiclass support vector machines. IEEE Trans. on Neural Networks, 2002, 13(2):415～425

11 UCI Repository of Machine Learning Databases. http:// www.ics.uci.edu/～mlearn /MLR Reposito ry.html