ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.4, April 2008, pp.993−1003
DOI: 10.3724/SP.J.1001.2008.00993

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于分布式学习的大规模网络入侵检测算法*

刘衍珩[1,2], 田大新[1,2+], 余雪岗[1,2], 王 健[1,2]

[1](吉林大学 计算机科学与技术学院,吉林 长春 130012)

[2](吉林大学 符号计算与知识工程教育部重点实验室,吉林 长春 130012)

Large-Scale Network Intrusion Detection Algorithm Based on Distributed Learning

LIU Yan-Heng[1,2], TIAN Da-Xin[1,2+], YU Xue-Gang[1,2], WANG Jian[1,2]

[1](College of Computer Science and Technology, Jilin University, Changchun 130012, China)

[2](Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

+ Corresponding author: Phn: +86-431-85168355, E-mail: daxin222@163.com, daxin222@email.jlu.edu.cn

Liu YH, Tian DX, Yu XG, Wang J. Large-Scale network intrusion detection algorithm based on distributed learning. *Journal of Software*, 2008,19(4):993−1003. http://www.jos.org.cn/1000-9825/19/993.htm

Abstract: As Internet bandwidth is increasing at an exponential rate, it's impossible to keep up with the speed of networks by just increasing the speed of processors. In addition, those complex intrusion detection methods also further add to the pressure on network intrusion detection system (NIDS) platforms, and then the continuous increasing speed and throughput of network pose new challenges to NIDS. In order to make NIDS effective in Gigabit Ethernet, the ideal policy is to use a load balancer to split the traffic and forward them to different detection sensors, and these sensors can analyze the splitting data in parallel. If the load balancer is required to make each slice containing all the necessary evidence to detect a specific attack, it has to be designed complicatedly and becomes a new bottleneck of NIDS. To simplify the load balancer, this paper puts forward a distributed neural network learning algorithm. By using the learning algorithm, a large data set can be split randomly and each slice data is handled by an independent neural network in parallel. The first experiment tests the algorithm's learning ability on the benchmark of circle-in-the-square and compares it with ARTMAP (adaptive resonance theory supervised predictive mapping) and BP (back propagation) neural network; the second experiment is performed on the KDD'99 Data Set which is a standard intrusion detection benchmark. Comparisons with other approaches on the same benchmark show that it can perform detection at a high detection speed and low false alarm rate.

Key words: intrusion detection system; network behavior; neural network; distributed learning

摘 要: 计算机网络的高速发展,使处理器的速度明显低于骨干网的传输速度,这使得传统的入侵检测方法无法应用于大规模网络的检测.目前,解决这一问题的有效办法是将海量数据分割成小块数据,由分布的处理节点并行处

* Supported by the National Natural Science Foundation of China under Grant No.60573128 (国家自然科学基金); the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20060183043 (国家教育部高校博士点基金)

Received 2007-03-29; Accepted 2007-06-14

理.这种分布式并行处理的难点是分割机制,为了不破坏数据的完整性,只有采用复杂的分割算法,这同时也使分割模块成为检测系统新的瓶颈.为了克服这个问题,提出了分布式神经网络学习算法,并将其用于大规模网络入侵检测.该算法的优点是,大数据集可被随机分割后分发给独立的神经网络进行并行学习,在降低分割算法复杂度的同时,保证学习结果的完整性.对该算法的测试实验首先采用基准测试数据 circle-in-the-square 测试了其学习能力,并与 ARTMAP(adaptive resonance theory supervised predictive mapping)和 BP(back propagation)神经网络进行了比较;然后采用标准的入侵检测测试数据集 KDD'99 Data Set 测试了其对大规模入侵的检测性能.通过与其他方法在相同数据集上的测试结果的比较表明,分布式学习算法同样具有较高的检测效率和较低的误报率.

关键词: 入侵检测系统;网络行为;神经网络;分布式学习

中图法分类号: TP393 **文献标识码:** A

随着互联网的普及和业务量的不断增长,网络的规模和传输速度也急剧增长,这使得如何在高速环境和海量数据中检测异常行为,成为入侵检测领域面临的又一大难题.传统的误用检测和异常检测技术无论在硬件设计还是在检测算法方面都难以满足对高速大规模网络进行有效检测的要求.在硬件方面,检测系统的CPU处理能力、内存容量等资源制约了检测速度,现有的高速入侵检测系统的实测速度也只能达到 10^3Mbit/s,这显然不能适用于骨干网 10Gbit/s甚至更高的速度.在检测算法中,误用检测多基于规则的查找匹配方法;异常检测多基于神经网络和数据挖掘等智能方法.随着网络攻击种类的不断增加,必然要求更大的规则库来进行匹配,而且数据量的增大也加剧了智能方法的运算量.上述问题必然导致消耗更大的计算资源.由于网络带宽的增长快于处理器速度的增长,因此,必须研究可行的技术和算法来解决高速网络环境下的入侵检测问题.

为了使入侵检测系统适应高速网络环境,一方面要在保证较低误报率的同时提高检测算法的效率;另一方面要采用分布式的体系结构进行数据采集并实现并行处理.为了提高算法的检测效率,文献[1–3]从提高数据包的抓取、过滤和匹配速度入手,使入侵检测系统能在高速网络环境下降低丢包率,从而提高检测的准确率;对于基于规则的检测方法,入侵规则条目的不断增长使匹配算法成为入侵检测系统的性能瓶颈,为此,文献[4,5]提出了对字符串匹配算法的改进,以增加检测系统的吞吐率.在异常检测方面,模式识别算法的效率是决定检测系统性能的关键,研究人员在将各种机器学习算法集成[6–8]以提高检测准确率的同时,采用降低样本数据维数的方法提高检测速度[8–10].上述方法能够提高基于主机和小规模局域网的入侵检测系统的效率和准确性,但检测系统处理速度和高速网络传输速度之间的差距,使上述方法无法适用于骨干网络的检测.为此,文献[11]提出了数据流的分片(分割)机制,其核心思想是将高速海量数据分割成不同部分并分配给分布的独立检测节点并行处理.这种分布式并行处理机制是目前解决检测系统处理速度无法匹配网络传输速度问题的最佳途径,但这种机制也给检测算法的设计与实现带来了难题.因为保证上述机制有效的前提是分配给独立检测节点的数据包含了某次入侵行为的全部或主要信息,而这对于以数据包为基本传输单位并具有"流"和"连接"特征的计算机网络来说,显然是一个棘手的问题.

实现分布式并行入侵检测系统的关键是数据的分割算法,一种理想的分割算法应满足以下条件:① 分配给每个处理节点的数据块大小相同;② 每个分片数据块中包含的信息能够使检测算法准确地检测出入侵行为;③ 分割方法简单、高效[12].在目前提出的分布式并行检测机制中,文献[12]采用的分片机制是将一次连接的数据包全部优先分配给队列长度较短的一个检测节点,对扫描或DoS类攻击由一个专用的检测节点检测,该方法优先考虑了条件②和条件③,但无法满足条件①.尽管采用将数据优先分配给数据队列较短的节点,但当一次连接的数据量很大时,必然会导致某些处理节点产生拥塞从而出现丢包现象.文献[13]在数据分配时综合考虑处理节点当前会话数、数据包数、CPU利用率和内存空间大小来决定优先分配到哪个节点.文献[14]在将数据分发给各个处理节点之前,采用提前筛选和局部缓存方法对数据进行预处理,以减轻处理节点的负担,这显然无法满足条件③,从而使负载均衡模块成为检测系统的瓶颈.为了提高分割算法的效率,文献[15]采用基于哈希的包分配机制以减轻负载均衡模块的负担,该方法虽然实现简单但无法有效满足条件②.

综上分析,现有的分割算法为了较好地满足条件②,不得不设计复杂的负载均衡模块,从而导致无法完全符

合条件①和条件③.同时,分割算法必然会在一定程度上破坏数据的完整性,因此,需要对现有的入侵检测算法,尤其是异常算法进行改进,以减少分布式并行处理产生的误差.尽管在数据挖掘、分类等领域提出了元学习[16,17]、协作贝叶斯学习[18]等并行学习算法,并将其应用到信用卡[19,20]、网络[21]等异常检测中,但上述方法的核心是对各个子数据集的学习结果采用投票(voting)、组合(combiner)、仲裁(arbiter)、SCANN(stacking with correspondence analysis and nearest neighbor)等机制进行合并.这种机制面临的一个难题是如何准确选择各个子数据集对应的偏置使学习算法对整个数据进行有效学习,而网络中传输信息的面向连接特征使该问题更加突出.

本文提出的随机分割方法的核心思想是,样本数据可以被随机分割成不相交的子集,不用在分割样本前计算出样本之间的关联性(如将具有相同的源/目的地址、相同的源/目的端口以及 TCP(transmission control protocol)序列号连续的数据包分割给同一个子处理节点),这样不仅降低了分割算法的计算复杂度,而且避免了某个子处理节点接收的样本过多而导致的漏报问题的发生.随机分割使分割算法完全满足条件①和条件③,但会导致异常检测方法产生较高的误报率,即无法满足条件②.因为样本被随机分割后,代表一个网络行为的完整样本集可能被分割成多个不相交子集,并提交到不同子处理节点.为了得到网络行为描述,目前主要采用投票、组合、仲裁、SCANN 等机制将各个子集的学习结果加以合并.尽管合并后的学习结果可以有效地提高系统的准确率,但在如下情况下会存在较大误差:当存在大量冗余且学习结果并不精确的子集时;当准确地预测或分类结果存在于某一子集中时.为了避免在上述两种情况下无法满足条件②的问题,本文提出基于神经网络的分布式学习算法,并在文献[22,23]的基础上将其应用于分布式网络行为学习.该算法的目标是对任意分割的网络数据都能进行有效的学习,进而降低对负载均衡模块的要求.

1 分布式神经网络学习算法

尽管每个独立的神经网络以并行方式处理数据,但让多个分布的神经网络合作处理一个任务是一个难点.因为神经网络的学习过程要求把所有的样本数据都提交给神经网络进行训练,直至其在一次或多次循环训练后稳定.这种机制使得当数据量非常大以至内存空间无法满足时,学习无法进行,而大数据集问题在实际应用中非常普遍.本文提出分布式神经网络学习算法让多个独立的神经网络同时处理随机分割的部分数据,并将所得知识通过集中学习加以合并,这样在发挥单个神经网络并行处理能力的同时,使其可以对分布存储的大规模数据集学习.

1.1 学习过程

学习算法的主要过程如下:

Step 1. 将大块数据集分割成小块,然后将小块数据提交给各个独立的神经网络;

Step 2. 各个神经网络对其分得的子数据集进行学习直至稳定;

Step 3. 利用各个神经网络的学习结果生成新的数据集,该数据集远小于各个子数据集之和;

Step 4. 对新的数据集进行学习直至稳定.

整个过程包括两个学习过程:分布学习和集中学习.分布学习的样本为子数据集,学习算法既要保证学习到子数据集中的完整知识,又不能丢弃因分割导致的部分知识;集中学习的样本为由各神经网络学习结果生成的新数据集,该数据集不仅包含各子数据集的知识,而且样本数量远小于原始数据集.一个稳定的神经网络学习到的知识存储在权重矩阵$\boldsymbol{W}_{(m\times n)}$中,其中,$m$为神经元的个数,$n$为每个神经元的维数.在上述学习过程中,每个分布学习的神经网络中神经元的维数n等于每个样本$\boldsymbol{x}_{(1\times n)}$的维数.当神经网络稳定后,其权重矩阵$\boldsymbol{W}$的每一行为从子数据集中学习得到的知识,将$\boldsymbol{W}$中的学习次数较少的知识点(有可能是部分知识点)与剩余的知识一起生成新的数据集.例如,原始数据集$\boldsymbol{X}$含有$p\times q$个样本,数据集$\boldsymbol{X}$被分割成p个子数据集${}^{(i)}\boldsymbol{X}(i=1,\ldots,p)$,每个子数据集含有$q$个样本.当对${}^{(i)}\boldsymbol{X}$进行学习的神经网络学习稳定后,其权重矩阵${}^{(i)}\boldsymbol{W}$含有${}^{(i)}r$行(${}^{(i)}r<<q$),由所有${}^{(i)}\boldsymbol{W}$中的知识点生成新的样本数据集$\tilde{\boldsymbol{X}}$远小于原始数据集$\boldsymbol{X}$.

学习次数较少的知识点是指在分布学习过程中训练次数较少的某些神经元,即权重矩阵中的某些行向量;

剩余的知识是训练次数较多的神经元.导致训练次数较少的原因,一方面可能是代表该知识点的样本数量本来就很少,另一方面的原因可能是因为代表某类知识点的样本被分割到不同子神经网络而导致单个子神经网络得到的样本数量少.第 2 种情况是导致分布式学习准确率低的主要原因,要克服该问题,就需要保留该类学习次数较少的知识点,并与学习次数足够多的知识点共同构成用于集中学习的新数据集.

上述学习过程可以通过基于 Hebb 规则类神经网络学习算法来实施是因为该类神经网络具有如下两个特征:① Hebb 学习是一种局部学习;② 该类神经网络的权重向量代表了知识点.这两个特征确保了即使代表某类知识的训练样本被分割到多个子集中,也能在分布学习时被保留并在集中学习后被抽取出来,从而避免了当存在大量冗余且学习结果并不精确的子集时,分割学习误差较大的问题.而其他学习算法因不能同时具备上述特征而无法采用上述学习过程对大规模数据加以学习.

1.2 风险界分析

本文提出的分布式学习算法在分布学习阶段采用的是局部风险最小化方法,为了解决局部风险最小化问题,Vapnik 定义的局部风险最小化模型为

$$R(\boldsymbol{w},\beta;\boldsymbol{x}_0)=\int L(y,f(\boldsymbol{x},\boldsymbol{w}))\frac{K(\boldsymbol{x},\boldsymbol{x}_0;\beta)}{\kappa(\boldsymbol{x}_0;\beta)}\mathrm{d}p(\boldsymbol{x},y) \tag{1}$$

其中,

$$\kappa(\boldsymbol{x}_0;\beta)=\int K(\boldsymbol{x},\boldsymbol{x}_0;\beta)\mathrm{d}p(\boldsymbol{x}) \tag{2}$$

$K(\boldsymbol{x},\boldsymbol{x}_0;\beta)$为邻域函数,$\beta$为界参数.

上述模型的目标是在函数集$f(\boldsymbol{x},\boldsymbol{w})$和点$\boldsymbol{x}_0$的不同邻域函数上最小化局部风险泛函(1).分布式学习算法不是在点$\boldsymbol{x}_0$上,而是在相应的神经元上最小化风险,因此,本文提出分布式学习下的局部风险最小化模型.

定义 1. 分布式风险最小化模型$R^d(\boldsymbol{w},\beta)$为

$$R^d(\boldsymbol{w},\beta)=\int L(y,f(\boldsymbol{x},\boldsymbol{w}))\frac{K(\boldsymbol{x},\boldsymbol{w};\beta)}{\kappa^d(\boldsymbol{w};\beta)}\mathrm{d}p(\boldsymbol{x},y) \tag{3}$$

其中,

$$\kappa^d(\boldsymbol{w};\beta)=\int K(\boldsymbol{x},\boldsymbol{w};\beta)\mathrm{d}p(\boldsymbol{x}) \tag{4}$$

模型(3)的经验风险泛函定义为

$$R^d_{emp}(\boldsymbol{w},\beta)=\frac{1}{\kappa^d(\boldsymbol{w};\beta)}\sum_{i=1}^{n}L(y,f(\boldsymbol{x}_i,\boldsymbol{w}))K(\boldsymbol{x}_i,\boldsymbol{w};\beta) \tag{5}$$

Vapnik 证明,经验风险最小化原则下对于函数集中的所有函数,如果 $0\le L(y,f(\boldsymbol{x},\boldsymbol{w}))\le B$,则下列关系至少以概率 $1-\eta$成立:

$$R(\boldsymbol{w})\le R_{emp}(\boldsymbol{w})+\frac{B\varepsilon(\ell)}{2}\left[1+\sqrt{1+\frac{4R_{emp}(\boldsymbol{w})}{B\varepsilon(\ell)}}\right] \tag{6}$$

其中,$\varepsilon(\ell)=4\dfrac{\left(\ln\dfrac{2\ell}{h}+1\right)h-\ln\eta/4}{\ell}$,$h$为VC维.尽管式(1)和式(3)两个模型在计算误差函数的具体比较点上有所不同,式(3)是在竞争获胜的神经元上,而式(1)是在点$\boldsymbol{x}_0$上,但都满足统计学习理论中式(6)描述的经验风险最小化原则.由式(6)可知,实际风险由$R_{emp}(\boldsymbol{w})$和$\dfrac{B\varepsilon(\ell)}{2}\left[1+\sqrt{1+\dfrac{4R_{emp}(\boldsymbol{w})}{B\varepsilon(\ell)}}\right]$两部分组成,第 1 部分为经验风险,第 2 部分为置信区间.通常,学习方法是首先通过选择模型来固定置信区间,然后通过最小化经验风险泛函来求最小风险.因为缺乏对置信区间的认识,这种选择往往依赖于先验知识和经验.为此,Vapnik 提出结构风险最小化原则,即选择最小经验风险与置信区间之和最小的子集,在这个子集中,使经验风险最小的函数为所求的最优函数.在分布式学习中,为了防止被分割的知识丢失,在分布学习阶段没有采用结构风险最小化原则,而在集中学习阶段

采用后修剪算法实现结构风险最小化.

1.3 完整性分析

分布式学习算法对随机分割的数据进行学习的结果如果等价于对完整数据的学习结果,则说明分布式学习算法具备完整性.从有限数据点恢复其背后隐含的函数 $f(\boldsymbol{x},\boldsymbol{w})$是一个反问题,因而往往是不适定的.为此,Tikhonov 提出了正则化方法解决不适定问题.正则化理论要求的最小化泛函为

$$R(\boldsymbol{w})=R_s(\boldsymbol{w})+\lambda R_c(\boldsymbol{w}) \tag{7}$$

其中,$R_s(\boldsymbol{w})$为实际风险项,$R_c(\boldsymbol{w})$为正则化项,λ为正则化参数.正则化的基本思想是通过某些含有解的先验知识的非负的辅助泛函来使解稳定.通过分析表明,本文提出的分布式学习方法与采用正则化理论的学习方法等价.

定理. 分布式学习算法等价于正则化方法.

证明:分布学习得到的权重矩阵${}^{(i)}\boldsymbol{W}$的每一个行向量${}^{(i)}\boldsymbol{W}_j$为一部分样本的邻域函数中心,因此,对于该部分样本中的一个样本${}^{(i)}\boldsymbol{X}_k$,

$${}^{(i)}\boldsymbol{W}_j={}^{(i)}\boldsymbol{X}_k+\boldsymbol{A}_k \tag{8}$$

在新数据集 ${}^{(i)}\tilde{\boldsymbol{X}}$ 中,

$${}^{(i)}\tilde{\boldsymbol{X}}_l={}^{(i)}\boldsymbol{W}_j+\boldsymbol{B}_m \tag{9}$$

其中,$|\boldsymbol{A}_{ki}|\le\beta,|\boldsymbol{B}_{mi}|\le\beta$,合并式(8)和式(9)得到:

$${}^{(i)}\tilde{\boldsymbol{X}}_l={}^{(i)}\boldsymbol{X}_k+\boldsymbol{A}_k+\boldsymbol{B}_m \tag{10}$$

对于局部风险最小化模型(3)中的误差函数 $L(y,f(\boldsymbol{x},\boldsymbol{w}))$,在新数据集 $\tilde{\boldsymbol{X}}$ 中为 $L(y,f(\boldsymbol{x}+\boldsymbol{a}+\boldsymbol{b},\boldsymbol{w}))$,将 $f(\boldsymbol{x}+\boldsymbol{a}+\boldsymbol{b},\boldsymbol{w})$按泰勒级数展开得到:

$$\begin{aligned} f(\boldsymbol{x}+\boldsymbol{a}+\boldsymbol{b},\boldsymbol{w}) &= f(\boldsymbol{x},\boldsymbol{w})+\nabla f(\boldsymbol{x}+\boldsymbol{a}+\boldsymbol{b},\boldsymbol{w})^T(\boldsymbol{a}+\boldsymbol{b})+ \\ &\quad \frac{1}{2}(\boldsymbol{a}+\boldsymbol{b})^T\nabla^2 f(\boldsymbol{x}+\boldsymbol{a}+\boldsymbol{b},\boldsymbol{w})(\boldsymbol{a}+\boldsymbol{b})+\ldots \\ &= f(\boldsymbol{x},\boldsymbol{w})+\Delta g(\boldsymbol{x}) \end{aligned} \tag{11}$$

其中,$|\boldsymbol{a}_i+\boldsymbol{b}_i|\le 2\beta$,$\nabla f(\boldsymbol{z})$为$n$元函数$f(\boldsymbol{z})=f(z_1,z_2,\ldots,z_n)$的梯度,$\nabla^2 f(\boldsymbol{z})$为赫森矩阵.$L(y,f(\boldsymbol{x},\boldsymbol{w}))$取最小二乘法,则

$$\begin{aligned} R(\boldsymbol{w}) &= \int (y-f(\boldsymbol{x}+\boldsymbol{a}+\boldsymbol{b},\boldsymbol{w}))^2 \mathrm{d}p(\boldsymbol{x},y) \\ &= \int (y-f(\boldsymbol{x},\boldsymbol{w}))^2 \mathrm{d}p(\boldsymbol{x},y)+\int ((y-f(\boldsymbol{x},\boldsymbol{w}))\Delta g(\boldsymbol{x})+\Delta g(\boldsymbol{x})^2)\mathrm{d}p(\boldsymbol{x},y) \end{aligned} \tag{12}$$

由式(12)可以发现,分布式学习即为在风险泛函 $\int (y-f(\boldsymbol{x},\boldsymbol{w}))^2 \mathrm{d}p(\boldsymbol{x},y)$ 的基础上增加惩罚项

$$\int ((y-f(\boldsymbol{x},\boldsymbol{w}))\Delta g(\boldsymbol{x})+\Delta g(\boldsymbol{x})^2)\mathrm{d}p(\boldsymbol{x},y).$$

这种方法在均衡神经网络的偏置与方差[24]中普遍采用,因此,在β控制在一定小的范围的情况下[25]等价于正则化方法. □

1.4 竞争学习算法

为了防止数据被分割后,代表某种行为的小规模数据被淹没,本文借鉴自适应谐振理论的调整子系统机制,在学习时以相似度评价为基础,并通过辨别样本的类别属性进行谐振,对竞争学习时无法识别的样本数据,通过增加神经元来代表该部分知识的中间学习结果.

根据 Hebbian 假设,可用能量函数表示一个神经元的学习规则:

$$E(\boldsymbol{w})=-\psi(\boldsymbol{w}^T\boldsymbol{x})+\frac{\alpha}{2}\|\boldsymbol{w}\|_2^2 \tag{13}$$

其中,$\boldsymbol{w}$ 是突触权值向量,$\boldsymbol{x}$ 是输入样本向量,$\psi()$为可微函数,$\alpha\ge 0$ 为遗忘系数.神经元的输出为

$$y=\frac{\mathrm{d}\psi(v)}{\mathrm{d}v}=f(v) \tag{14}$$

其中,$v=\boldsymbol{w}^T\boldsymbol{x}$是神经元的活跃系数.通过快速下降法导出连续时间的学习规则:

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t}=-\mu\nabla_w E(\boldsymbol{w}) \tag{15}$$

其中,$\mu>0$ 为学习速度系数,$\nabla_w E(\boldsymbol{w})=\partial(\boldsymbol{w})/\partial\boldsymbol{w}$,则公式(13)的梯度为

$$\nabla_w E(\boldsymbol{w})=-f(v)\frac{\partial v}{\partial \boldsymbol{w}}+\alpha\boldsymbol{w}=-y\boldsymbol{x}+\alpha\boldsymbol{w} \tag{16}$$

由此可得单神经元的学习规则为

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t}=\mu[y\boldsymbol{x}-\alpha\boldsymbol{w}] \tag{17}$$

则离散时间的学习规则为

$$\boldsymbol{w}(t+1)=\boldsymbol{w}(t)+\mu[y(t+1)\boldsymbol{x}(t+1)-\alpha\boldsymbol{w}(t)] \tag{18}$$

相似度 d 按 Minkowski 度量计算:

$$d_p(\boldsymbol{x},\boldsymbol{y})=\left(\sum_{i=1}^{l} w_i\mid x_i-y_i\mid^p\right)^{1/p} \tag{19}$$

其中,x_i和y_i是向量$\boldsymbol{x}$和$\boldsymbol{y}$的第i个元素,$i=1,\ldots,l$;$w_i\geq 0$ 是权重系数.

若遗忘系数$\alpha=1$,则神经元活跃时($y=1$),由式(18)可得第 i 个神经元的学习规则为

$$\boldsymbol{w}_i(t+1)=\boldsymbol{w}_i(t)+\mu\times\gamma_i[\boldsymbol{x}(t+1)-\boldsymbol{w}_i(t)] \tag{20}$$

为了防止因数据分割而使某些网络行为模式无法被学习到,定义相似度的门限值ϑ,对一个输入样本$\boldsymbol{x}$,若竞争获胜神经元j的相似度$d_j\leq\vartheta$,则获胜神经元的γ_j为 1;若$d_j>\vartheta$,则添加新的神经元并将其突触权值置为$\boldsymbol{x}$.

1.4.1 后修剪算法

对分割数据的学习结束后,各个分割数据的学习结果中包含了一些较完整的网络行为模式和一些未完全学习到并被分割表示的网络行为模式.因此,需要对这些中间结果构成的样本空间进行再学习,从而形成完整的网络行为知识.上述过程在解决了网络行为的一些特征有可能因为被分割而无法形成知识的问题的同时,也降低了学习结果的泛化能力,因为样本数据中一些特征有可能被重复表示从而导致过学习.为此,采用后修剪算法来实现结构风险最小化原则.后修剪算法以训练后的每一个神经元为修剪的候选对象,将相似的神经元合并为一个神经元或多个(总数比合并前要少)神经元.

具体修剪算法如下:

Step 0. 如果旧神经元矩阵(*oldW*)为空,则修剪结束,否则继续;

Step 1. 计算旧神经元矩阵中第 1 个神经元(*fw*)与其他神经元的相似度;

Step 2. 查找与 *fw* 最相近的神经元(*sw*);

Step 3. 如果 *sw* 与 *fw* 之间的距离大于修剪门限值,则将 *fw* 加入新神经元矩阵(*newW*),同时将 *fw* 从 *oldW* 中删除并返回 Step 0;否则继续;

Step 4. 分别将 *fw* 的训练次数和 *sw* 的训练次数赋给变量 *ft* 和 *st*;

Step 5. 计算新神经元(*nw*)和 *nw* 的训练次数(*nt*),$nw=(fw\times ft+sw\times st)/(ft+st)$,$nt=ft+st$;

Step 6. 将 *fw* 和 *sw* 从 *oldW* 中删除,将 *nw* 添加到 *newW* 中,返回 Step 0.

1.4.2 分布式学习算法

学习算法的主体学习过程可描述如下:

Step 0. 初始化学习速度系数μ,相似度门限值ϑ;

Step 1. 接收第 1 个样本向量$\boldsymbol{x}$,添加第 1 个神经元$\boldsymbol{w}_0$并置初始值为$\boldsymbol{x}$;

Step 2. 判断学习是否结束:若否,则从样本空间中接收一个样本向量,并按公式(19)计算相似度值d_i;

Step 3. 由竞争函数判断获胜的神经元j,若$d_j>\vartheta$,则添加新的神经元,并使其突触权值为$\boldsymbol{x}$,返回Step 2,否则继续;

Step 4. 按公式(20)更新突触权值,返回 Step 2.

在分布学习阶段,将分割后的数据提交给各个子神经网络按上述学习过程进行训练,训练结束后,由各个子

神经网络的权重矩阵生成新的数据集;在集中学习阶段,对新数据集进行学习并在学习稳定后进行修剪,最终得到学习结果.

2 实 验

实验测试首先采用 circle-in-the-square 基准测试数据集测试了算法的学习性能,然后用第 3 届国际知识发现和数据挖掘竞赛(KDD'99)的基准测试数据集测试了算法对大规模数据的入侵检测能力.

2.1 Circle-in-the-Square基准测试

Circle-in-the-Square 问题要求神经网络能够准确分辨出一单位正方形的点中位于一个圆内和圆外的点,该圆位于正方形中且面积为单位正方形的一半.本文实验使用文献[26]的基准测试数据,该数据集可以从 CELEST Technology Website 下载.数据集 cis_train2.txt 中包含 1 000 个样本数据,为了测试分布式学习算法的学习性能,将数据集按图 1 进行分割,其中,*A*1,*A*2 组成 *A* 块数据集,*B*1,*B*2 组成 *B* 块数据集.对该样本的随机分割方法有很多种,如将样本从 1~1 000 标号,然后按顺序从前往后分割成几个不相交的子集,或通过生成随机数的方法将这 1 000 个样本分割成几个不相交的子集等等.针对该 Benchmark,这种分割是一种随机分割,因为该 Benchmark 要解决的问题是分辨出一单位正方形的点中位于一个圆内和圆外的点,而分割到各个子神经网络的样本数据中不仅包含圆内的点和圆外的点,而且在这些分割后的 3 个样本子集中,分别是由一块完整子样本、两块相邻的子样本和两块不相邻的子样本数构成,所以,这样分割是一种随机分割.

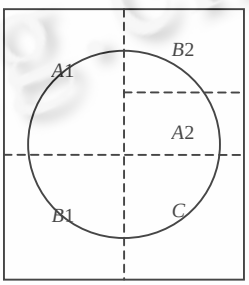


Fig.1 The distribution of the slice data

图 1 分割数据的分布情况

当相似度门限值 0.05 增加到 0.08 时,神经网络的训练次数、神经元数和准确率与其他方法的比较列在表 1 中.对于分布式学习,其训练次数为分布学习中训练次数最多的神经网络的训练次数加上集中学习的训练次数.在训练的一个周期中,若谐振发生,某个样本可能被多次学习,所以,分布式学习的训练次数等价于其他方法的训练周期乘以样本数.

Table 1 The testing result of circle-in-the-square

表 1 Circle-in-the-Square 测试结果

	Distributed learning	ARTMAP	BP
Number of training	518~621	1×100~1×100000	5000×150~5000×14000
Accuracy rate (%)	96~98	88.6~98	90
Number of neuron	91~196	12~121	21~401

2.2 KDD CUP'99

1998 年,MIT Lincoln 实验室与 DARPA 合作开展了入侵检测系统评估,该计划的任务之一是提供包括主机日志和网络流量在内的用于入侵检测的数据集.KDD CUP'99 对 DARPA 提供的 9 周 tcpdump 数据进行了适当处理和特征提取,得到大约 700 万条连接记录,其中包含大量攻击记录.完整的训练数据集包含 4 999 000 条连接记录,每条记录由 42 个属性组成,其中最后一个属性描述该条记录是正常连接还是某种入侵行为.这些入侵行为被分成 4 大类:probe,denial of service (DOS),user-to-root (U2R)和 remote-to-local (R2L).本文以完整训练数据 10%的数据子集作为训练数据,该数据集共 494 019 条记录,其中,正常行为记录 97 276 条,入侵行为记录 396 743 条,入侵行为共 22 种;测试数据使用带标识的测试数据集,该数据集共 311 029 条记录,其中,正常行为记录 60 593 条,入侵行为记录 250 436 条,入侵行为共 37 种,这些入侵行为中包含训练数据中的 20 种入侵行为和 17 种训练数据中未出现的入侵行为.

2.2.1 数据处理

KDD Cup'99 数据在每条记录的 42 个属性中,除最后一个以外,共有 7 种符号变量和 34 种连续变量.如flag属性为符号变量共有 11 种标识:SH-1,RSTR-2,OTH-3,S0-4,REJ-5,S2-6,SF-7,S1-8,S3-9,RSTOS0-10,RSTO-11.在本文中,符号变量值用矩阵$\boldsymbol{A}$表示,含有g种标识的符号变量$\boldsymbol{A}_i$定义为

$$\boldsymbol{A}_i=\{A_i^1,A_i^2,...,A_i^g\} \tag{21}$$

连续变量用向量$\boldsymbol{b}_{1\times n}$表示,即含有$n$个连续变量.输入变量$\boldsymbol{X}$定义为

$$\boldsymbol{X}=\{\boldsymbol{A}_1,\ldots,\boldsymbol{A}_m,b_1,\ldots,b_{n-m}\} \tag{22}$$

即输入变量共有 n 个属性,其中包含 m 种符号变量.

对于连续变量Minkowski度量公式中的$|x_i-y_i|$,可在归一化后直接计算,但对于符号变量无法直接计算.本文采用类似Hamming距离的方法计算.对于符号变量,y_i不是一个数量值,而是突触权值(神经元)N对应符号属性$\boldsymbol{A}_i$的一个标记,如果 $x_i=A_i^k(k\in 1,...,g)$, 则

$$|x_i-y_i|=1-\frac{c_k}{C} \tag{23}$$

其中,c_k为N在学习过的样本中 A_i^k 出现过的次数:

$$c_k=num(A_i^k) \tag{24}$$

C 为 N 的学习次数,即

$$C=\sum_{i=1}^{m}\sum_{j=1}^{g}num(A_i^j) \tag{25}$$

N学习 1 次,其对应的c_k值将加 1.

按上述方法计算符号变量的Minkowski度量,若一个训练样本的符号变量$\boldsymbol{A}_i$的标识值为 A_i^k, 则学习过 A_i^k 的次数越多的神经元N与该样本越相近,同时,该方法实现了归一化.

2.2.2 评测方法

入侵检测系统的检测结果记录格式见表 2,其中,错误报警分为正的误报(不是入侵误报为入侵,false positive,简称 FP)和负的误报(是入侵却漏报了,false negative,简称 FN);准确判断也分为正的(入侵行为判断为入侵,true positive,简称 TP)和负的(正常行为判断为正常,true negative,简称 TN).

Table 2 The recording format of detection result

表 2 检测结果的记录格式

		Detection result				
		Normal	Intrusion 1	Intrusion 2	…	Intrusion n
Actual behaviors	Normal	$TN00$	$FP01$	$FP02$	…	$FP0n$
	Intrusion 1	$FN10$	$TP11$	$FP12$	…	$FP1n$
	Intrusion 2	$FN20$	$FP21$	$TP22$	…	$FP2n$
	…	…	…	…	…	…
	Intrusion n	$FNn0$	$FPn1$	$FPn2$	…	$TPnn$

定义 2. 某种行为i的正确检测率 $TR=T_{ii}\Big/\sum_{j=1}^{n}R_{ij}$, 其中,$T_{ii}$为表 2 中行为$i$所在行和列的相交点对应的值,$R_{ij}$为表 1 中行为 i 所在行中行为 j 所在列对应的值.

定义 3. 某种行为i的正确预测率 $PR=T_{ii}\Big/\sum_{j=1}^{n}R_{ji}$, 其中,$T_{ii}$为表 2 中行为$i$所在行和列的相交点对应的值,$R_{ji}$为表 2 中行为 i 所在列中行为 j 所在行对应的值.

定义 4. 入侵检测系统的检测率 DR(detection rate)为检测出的入侵行为数与所有入侵行为数之比,若将表 2 中记录的数据看作$(n+1)\times(n+1)$距阵 $\boldsymbol{R}$,则: $DR=\sum_{i=1}^{n}\sum_{j=1}^{n}R_{ij}\Big/\sum_{i=0}^{n}\sum_{j=0}^{n}R_{ij}$.

定义 5. 入侵检测系统正的误报率 FPR(false positive rate)为正常连接被报告为入侵的数量与所有正常连

接之比,若按表 2 中的记录表示, $FPR=\sum_{i=1}^{n}FP0i\Big/\left(\sum_{i=1}^{n}FP0i+TN00\right)$.

2.2.3 学习结果

为了验证分布式学习算法的性能,首先将 494 019 条训练数据分成 50 份,按照数据集中记录的顺序每 10 000 条记录划分为 1 份,最后一块数据含 4 019 条记录.每块数据分别单独进行学习,分布式学习算法用 MATLAB 编写,速度系数μ=0.1,相似度门限值ϑ=0.1.学习结果如图 2 和图 3 所示,图中横坐标表示第几块数据,图 2 纵坐标表示神经元数,图 3 纵坐标表示行为种类数.

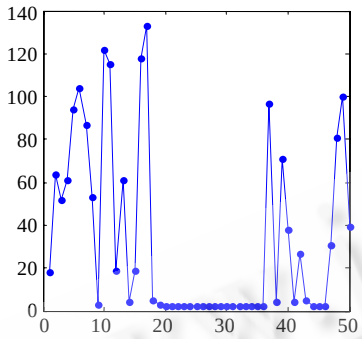


Fig.2 The number of neurons of each corresponding slices

图 2 每个数据块训练后对应的神经元数

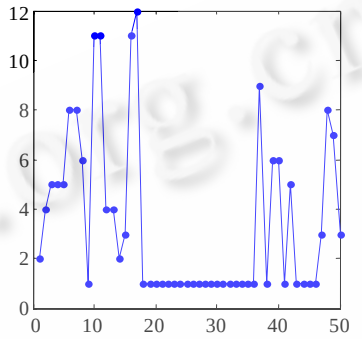


Fig.3 The number of behaviors of each corresponding slice

图 3 每个数据块训练后包含的行为种类数

由图 2 和图 3 可以看出,神经元的规模和行为种类的分布基本一致,学习结果较好地涵盖了数据中的规律,其中,18~36 块数据和 43~46 块数据中全部为 smurf 攻击的记录,因此,行为种类数为 1,对应的神经元为 2~5 个.在分布学习结果的基础上进行再学习,稳定后知识保存在 479 个神经元中.

2.2.4 检测结果

在测试数据集中共有 37 种入侵行为,首先,将它们划分到不同的大类中:

Probe: {portsweep,mscan,saint,satan,ipsweep,nmap}

DOS: {udpstorm,smurf,pod,land,processtable,warezmaster,apache2,mailbomb,Neptune,back,teardrop}

U2R: {httptunnel,ftp_write,sqlattack,xterm,multihop,buffer_overflow,perl,loadmodule,rootkit,ps}

R2L: {guess_passwd,phf,snmpguess,named,imap,snmpgetattack,xlock,sendmail,xsnoop,worm}

然后,提交给训练好的系统进行测试,检测结果见表 3.

Table 3 The detection result based on the benchmark

表 3 基于测试数据集的检测结果

		Detection result					
		Normal	Probe	DOS	U2R	R2L	TR (%)
Actual behaviors	Normal	58 273	376	232	810	902	96.2
	Probe	334	3 317	304	3	207	80.9
	DOS	5 974	576	223 588	755	560	96.6
	U2R	175	26	20	24	4	9.6
	R2L	14230	5	5	185	141	0.1
	PR (%)	73.8	77.1	99.7	1.4	7.8	

将检测结果与KDD'99 竞赛获胜团队的结果作比较,本文所用分布式学习算法对Normal,Probe和DOS的检测率和预测率与获胜方法基本相同,但对U2R和R2L的检测率较低.其他检测方法对U2R和R2L的检测率也普遍偏低,主要原因是上述两种行为与其他行为相比样本数量偏少,故无法有效形成稳定的知识,从而多数被误判为正常行为.分布式学习方法、KDD'99 竞赛前两名及其他检测方法[27]的DR和FPR结果列于表 4.

Table 4 Comparison with other approaches

表 4 不同算法检测性能的比较

Performance / Algorithm	Detection rate (%)	False positive rate (%)
Winning entry	91.9	0.5
Second place	91.5	0.6
Best linear GP—FP rate	89.4	0.7
Best GEdIDS—FP rate	91	0.4
Distributed learning	91.7	3.2

3 结 论

网络规模和带宽的不断增长,加剧了集中式入侵检测系统的丢包和高误报率.分布式处理为上述问题提供了到目前为止最为理想的解决方案.本文提出了基于分布式学习的入侵检测算法:将大规模数据随机分成小块数据,然后分发给独立的神经网络进行学习,最后通过集中学习得到最终结果.为了克服知识因为被分割而导致丢失的问题,本文提出了分布式风险最小化模型$R^d(w,\beta)$,并证明了分布式学习方法等价于正则化原则.在基准测试集circle-in-the-square上进行了实验,首先将 1 000 个样本分割成 3 部分,实验表明,对分割后的知识算法也能进行有效的学习.在对KDD'99 基准测试数据集的实验中,分布式学习算法将 494 019 条数据分成 50 份并行学习,学习过程最慢的神经网络用了 29 分钟学习结束,加上后期学习时间,总共用了 53 分钟完成学习.用一个神经网络对完整数据进行学习用了 5 小时 43 分钟.KDD Cup'99 中的获胜方法MP13 用PERGAMENT software运行了 6 个小时完成全部计算.在学习结束后的检测实验中,采用不同于训练数据的测试数据集,该集合中含有 17 种训练数据中未出现的入侵行为.分布式学习对各种行为:Normal,Probe,DOS的检测效果与其他方法不相上下,但对U2R和R2L的检测率比较低.其他检测方法对U2R和R2L的检测率也普遍偏低,主要原因是上述两种行为与其他行为相比样本数量偏少,所以无法有效形成稳定的知识,分布式学习更加剧了这一问题.通过对数据集以及学习和检测过程的跟踪分析,解决这一问题的有效办法是对属性加权.这个问题和无监督学习下对未知攻击的检测是我们下一步的研究重点.

References:

[1] Song H, Lockwood JW. Efficient packet classification for network intrusion detection using FPGA. In: Wilton S, ed. Proc. of the 13th Int'l Symp. on Field-Programmable Gate Arrays. New York: ACM Press, 2005. 238−245.

[2] Baker. ZK, Prasanna VK. A methodology for synthesis of efficient intrusion detection systems on FPGAs. In: Pocek KL, ed. Proc. of the 12th Annual IEEE Symp. on Field-Programmable Custom Computing Machines. Washington: IEEE Computer Society, 2004. 135−144.

[3] Tian DX, Liu YH, Li YL, Tang Y. Fast matching algorithm and conflict detection for packet filter rules. Journal of Computer Research and Development, 2005,42(7):1128−1135 (in Chinese with English abstract).

[4] Tuck N, Sherwood T, Calder B, Varghese G. Deterministic memory-efficient string matching algorithms for intrusion detection. In: Li VOK, ed. Proc. of the 23rd Conf. of the IEEE Communications Society. Piscataway: IEEE Press, 2004. 2628−2639.

[5] Tan L, Sherwood T. A high throughput string matching architecture for intrusion detection and prevention. In: Hill MD, ed. Proc. of the 32nd Int'l Symp. on Computer Architecture. Washington: IEEE Computer Society, 2005. 112−122.

[6] Mukkamala S, Sung AH, Abraham A. Intrusion detection using an ensemble of intelligent paradigms. Journal of Network and Computer Applications, 2005,28(2):167−182.

[7] Lee H, Chung Y, Park D. An adaptive intrusion detection algorithm based on clustering and kernel-method. In: Ng WK, ed. Proc. of the 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Berlin, Heidelberg: Springer-Verlag, 2006. 603−610.

[8] Xu X, Wang X. An adaptive network intrusion detection method based on PCA and support vector machines. In: Li X, ed. Proc. of the 1st Int'l Conf. on Advanced Data Mining and Applications. Berlin, Heidelberg: Springer-Verlag, 2005. 696−703.

[9] Aggarwal CC, Yu PS. An effective and efficient algorithm for high-dimensional outlier detection. The Int'l Journal on Very Large Data Bases, 2005,14(2):211−221.

[10] Rawat S, Pujari AK, Gulati VP. On the use of singular value decomposition for a fast intrusion detection system. Electronic Notes in Theoretical Computer Science, 2006,142(3):215−228.

[11] Kruegel C, Valeur F, Vigna G, Kemmerer R. Stateful intrusion detection for high-speed networks. In: Abadi M, ed. Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 2002. 285−294.

[12] Lai H, Cai S, Huang H, Xie J, Li H. A parallel intrusion detection system for high-speed networks. In: Jakobsson M, ed. Proc. of the Applied Cryptography and Network Security: 2nd Int'l Conf. Berlin, Heidelberg: Springer-Verlag, 2004. 439−451.

[13] Jiang W, Song H, Dai Y. Real-Time intrusion detection for high-speed networks. Computers & Security, 2005,24(4):287−294.

[14] Charitakis I, Anagnostakis KG, Markatos E. An active traffic splitter architecture for intrusion detection. In: Kotsis G, ed. Proc. of the 11th IEEE/ACM Int'l Symp. on Modeling, Analysis and Simulation of Computer Telecommunications Systems. Washington: IEEE Computer Society, 2003. 238−241.

[15] Schaelicke L, Wheeler K, Freeland C. SPANIDS: A scalable network intrusion detection loadbalancer. In: Valero M, ed. Proc. of the 2nd Conf. on Computing Frontiers. New York: ACM Press, 2005. 315−322.

[16] Giraud-Carrier C, Vilalta R, Brazdil P. Introduction to the special issue on meta-learning. Machine Learning, 2004,54(3):187−193.

[17] Fan W, Wang H, Yu P, Stolfo S. A framework for scalable cost-sensitive learning based on combing probabilities and benefits. In: Grossman RL, ed. Proc. of the 2nd SIAM Int'l Conf. on Data Mining. Philadelphia: SIAM Press, 2002. 437−453.

[18] Yamanishi K. Distributed cooperative Bayesian learning strategies. In: Freund Y, ed. Proc. of the 10th Annual Conf. on Computational Learning Theory. New York: ACM Press, 1997. 250−262.

[19] Chan PK, Stolfo SJ. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In: Rakesh A, ed. Proc. of the 4th Int'l Conf. on Knowledge Discovery and Data Mining. Menlo Park: AAAI Press, 1998. 164−168.

[20] Chan PK, Fan W, Prodromidis AL, Stolfo SJ. Distributed data mining in credit card fraud detection. IEEE Intelligent Systems, 1999, 14(6):67−74.

[21] Sung B, Jerzy B. A decision tree algorithm for distributed data mining: Towards network intrusion detection. LNCS 3046, Berlin, Heidelberg: Springer-Verlag, 2004. 206−212.

[22] Liu YH, Tian DX, Wang AM. ANNIDS: Intrusion detection system based on artificial neural network. In: Cloete I, ed. Proc. of the 2nd Int'l Conf. on Machine Learning and Cybernetics. Washington: IEEE Computer Society, 2003. 1337−1342.

[23] Tian DX, Liu YH, Wei D. ARTNIDS: A network intrusion detection system based on adaptive resonance theory. Chinese Journal of Computers, 2005,28(11):1882−1889 (in Chinese with English abstract).

[24] Geman S, Bienenstock E, Doursat R. Neural networks and the bias/variance dilemma. Neural Computation, 1992,4(1):1−58.

[25] Bishop CM. Training with noise is equivalent to Tikhonov regularization. Neural Computation, 1995,7(11):108−115.

[26] Carpenter GA, Grossberg S, Markuzon N, Reynolds JH, Rosen DB. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. on Neural Networks, 1992,3(5):698−713.

[27] Folino G, Pizzuti C, Spezzano G. GP ensemble for distributed intrusion detection systems. In: Singh S, ed. Proc. of the 3rd Int'l Conf. on Advanced in Pattern Recognition. Berlin, Heidelberg: Springer-Verlag, 2005. 54−62.

附中文参考文献:

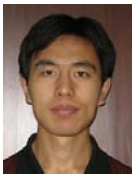
[3] 田大新,刘衍珩,李永丽,唐怡.数据包过滤规则的快速匹配算法和冲突检测.计算机研究与发展,2005,42(7):1128−1135.

[23] 田大新,刘衍珩,魏达.ARTNIDS:基于自适应谐振理论的网络入侵检测系统.计算机学报,2005,28(11):1882−1889.

刘衍珩(1958－),男,吉林松原人,博士,教授,博士生导师,主要研究领域为计算机网络安全,网络管理,移动 IP 和 QoS,传感器网络.

余雪岗(1974－),男,博士,讲师,主要研究领域为移动网络服务质量.

田大新(1980－),男,博士,主要研究领域为计算机网络安全,机器学习,人工神经网络.



王健(1981－),男,博士生,主要研究领域为计算机网络安全,复杂网络.