一种基于改进流形学习方法的云计算入侵检测模型

陈丹伟 侯 楠 孙国梓

(南京邮电大学计算机技术研究所 南京 210003) (南京邮电大学计算机学院 南京 210003)

摘 要 基于互联网的超级计算模式云计算引起了人们极大的关注,也面临着越来越多的安全威胁。主要构建能够适应云计算环境的入侵检测系统框架。将非线性流形学习算法引入本课题提出的模型,作为特征提取模块对云计算环境下采集的网络数据进行预处理;给出经典流形学习算法LLE的改进研究,以提高后续分类性能。实验表明,该算法是可行和高效的。

关键词 云计算,入侵检测,流形学习

中图法分类号 TP393 **文献标识码** A

Novel Cloud Computing Intrusion Detection Model Based on Improved Manifold Learning

CHEN Dan-wei HOU Nan SUN Guo-zi

(Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

(College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract The cloud computing technology which is a new Internet-based super-computing model has aroused great concern, facing an increasing number of security threats at the same time. This paper built a intrusion detection system framework which is adapt to the cloud computing environment proposed in this paper. The nonlinear manifold learning algorithm was introduced to the model proposed in this thesis to be the core algorithm of the data pre-processing module, and the improved LLE algorithm was given in order to improve the classification performance. The simulation result shows that the algorithm is effective and efficient.

Keywords Cloud computing, Intrusion detection, Manifold learning

云计算[1]是一种基于互联网的超级计算模式,它把存储在大量分布式计算机产品中的大量数据和处理器资源整合在一起协同工作。作为一种新兴的共享基础架构的方法,它可以将巨大的系统池连接在一起,以提供各种IT服务。这使得企业能够将资源切换到需要的应用上,根据需求访问计算机和存储系统。

云计算环境中强大的计算资源和巨大的存储能力对入侵者具有很大的诱惑力,成为入侵者的攻击目标。云计算的本质是虚拟化,无论是计算资源还是存储资源,都是实现各种资源的统计复用,为用户提供开放的用户接口,按需分配[2]。作为保障云计算环境安全的入侵检测系统,通过调度中心动态可靠地组织起来,形成安全保护的第一屏障[3]。由于云计算环境计算能力的优越性,对计算要求较高的各种算法,如流形学习算法,在云计算环境下都可以得到较好的实现,而流形学习算法对于提高入侵检测效率是极为有效的。在此背景下,本文主要构建能够适应云计算环境的分布式入侵检测系统框架。由于现实实验条件所限,本文并不针对整个系统模型进行试验分析,而主要着眼于基于虚拟组织的入侵检测系统数据预处理模块设计,并将改进的非线性流形学习算法引入本课题提出的模型,作为特征提取模块对云计算环境下采集的网络数据进行预处理,以提高后续分类性能。仿真实验证明本文提出的方法是可行和高效的。

1 云计算入侵检测模型CIDS

1.1 模型框架设计

本文提出的云计算入侵检测模型CIDS(Cloud Computing Intrusion Detecting System)是以虚拟组织为单位的分布式检测模型。虚拟组织是一个动态集合,它包含多个围绕一定资源共享规则和限定条件定义的个体或机构。因此,可以将虚拟组织看作是云计算环境整体的一个细胞。完成了细胞的功能以及细胞间的交互,就完成了云计算环境整体的入侵检测功能[4]。本文提出的云计算入侵检测模型由调度模块及目录服务器、数据采集预处理模块、数据分析模块、关联分析模块和容侵恢复模块等组成,计算云为系统提供工作环境。在该系统中,数据分析模块是一种资源,其所提供的服务就是检测算法,需要在目录服务器中进行注册才可能被使用。数

到稿日期:2010-02-05 返修日期:2010-04-07 本文受国家十一五科技支撑计划项目(2007BAK34B06),国家自然科学基金(60703086),南京邮电大学攀登计划项目(NY208009)资助。

陈丹伟(1970-),男,博士,副教授,硕士生导师,主要研究方向为计算机通信网与安全、嵌入式系统及应用等,E-mail:chendw@njupt.edu.cn;侯 楠(1984-),男,硕士生,主要研究方向为计算机通信网与安全;孙国梓(1972-),男,博士,副教授,硕士生导师,主要研究方向为计算机通信网与安全、计算机取证等。

据采集预处理模块相当于云中发起资源请求的用户，采集网络数据流进行预处理操作，将生成的待处理文件提交给数据分析模块。调度模块担当资源管理器的角色，负责进行入侵检测请求和检测资源之间的匹配，并根据一定的策略选择满足要求的资源，同时对资源进行管理，实现该入侵检测系统的可扩展性。容侵恢复模块与调度模块合作来处理分析模块性能下降或失效时，选择其它分析器继续分析任务的问题，以确保整个系统的鲁棒性。当数据采集预处理模块采集到网络数据流后，给调度模块提交请求。调度模块首先根据资源需求描述信息在资源信息数据库中查找到提供相应服务的资源，然后根据一定的资源分配算法确定为其提供服务的数据分析模块，并将相关信息返回给数据采集预处理模块。数据采集预处理模块将待处理文件传输给对应的数据分析模块进行处理，生成的告警信息传输给关联分析模块进行告警融合。同时容侵恢复模块对待处理文件进行备份，当相应数据分析模块性能下降或失效时，系统会自动根据目录服务器和调度模块选择合适的分析模块，继续分析未完成的任务。系统的体系结构如图 1 所示。

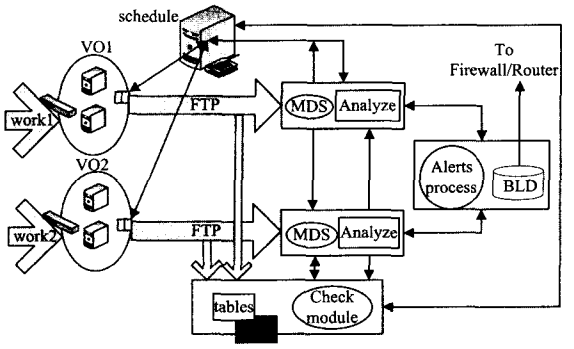

图 1　CIDS 系统体系结构

1.2　数据预处理模块设计

该模块负责收集虚拟组织的网络数据流，生成待处理文件，主要完成数据采集、数据预处理和通信传输功能。其基本架构如图 2 所示。

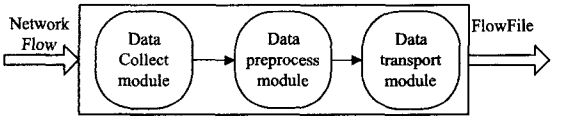

图 2　预处理模块架构

数据预处理模块是由大量计算机组成并且抽象而成的计算资源池，具有强大的存储和数据处理计算能力，可以充分利用云计算的优点来为入侵检测服务。该模块又细分成 3 个部分：第一部分完成原始数据数值化和归一化处理，经其处理后得到的是比较通用的数据；第二部分采用主流的降维方法对通用数据进行降维，从而提高分析模块的分析效率；最后一部分将前面模块输出的数据再转化为具体的适合分析模块中分类器特定格式的数据。这样设计的目的是使整个入侵检测模型具有更好的通用性，比如尝试在分析模块中使用不同的软件包，或要在数据预处理中应用更加复杂的其他处理，只需对相应模块作适当调整即可。

本文主要研究采用非线性流形学习方法对通用数据进行降维。入侵检测数据具有高维度、非线性等特点，经过前期处理后得到的向量组维数仍然较高，存在大量的对于入侵检测无关的冗余信息，这不但增加了分析模块的负担，影响了其执行效率，而且会降低整个系统检测的准确率，成为系统性能提高的瓶颈。前人在入侵检测领域的数据降维模块主要采用基于核的独立主成分分析等线性方法的变形，但又遇到了核函数选取等问题[6]，目前国际上还没有权威的解决方法。而作为新型学习方法的流形学习，对于网络流量等高维非线性数据的维度削减具有独特的优势，其在图像处理等领域中已取得相当好的效果[7]。虽然其对计算能力要求较高，但云计算环境强大的计算和存储能力完全可以满足其要求[8]，故在降维部分本文采用非线性流形学习算法来实现向量特征维数的削减。主要的流行学习方法包括 Isomap 和 LLE 等，由于复杂度相对较小且降维效果明显，本文采用局部线性嵌入 LLE 算法作为该部分的算法。

2　LLE 算法存在的问题及改进

流形学习旨在发现高维数据集分布的内在规律性，其基本思想是高维观测空间中的点由少数独立变量的共同作用在观测空间张成一个流形，如果能有效地展开观测空间卷曲的流形或发现内在的主要变量，就可以对该数据集进行降维。流形是微分几何学的基础，本质上是局部可坐标化的拓扑空间，可以看作是欧氏空间的非线性推广。LLE 算法认为在局部意义下数据的结构是线性的，或者说局部意义下的点在一个超平面上，因此任取一点，可以使用它的邻近点的线性组合来表示。经典 LLE 算法的步骤在文献[9]中有详细描述。

经典的 LLE 算法适用于非闭合的局部线性的学习，要求对样本采样充分平滑，特别是对噪音比较敏感，其中的一个重要参数——近邻个数 K 的选取显得特别重要。一般情况下，LLE 假设采集样本集分布比较均匀且连续，但通常由于网络流量的不均衡性，一段时间内网络流量有可能较大，而一段时间内网络流量有可能又较小，整体分布不均匀。在样本分布稀疏的区域里，K 个近邻所组成的局部邻域显然要比在样本分布比较密集的那部分区域内由 K 个近邻所组成的局部邻域大得多，而采用 ε 半径近邻选择又过于死板，所以本文采纳前两种方法的优点，提出了一种新型近邻个数 K 的选取方法。

在本文的方法中，近邻个数分为 K_{base} 和 K_{add} 两个部分，即 $K=K_{base}+K_{add}$，其中 K_{base} 表示基本值，K_{add} 表示附加值。首先根据经验给定 K_{base} 的值，然后计算样本点如 X_i 与它的第 K_{base} 个近邻的距离并与一个常数 h 相乘。将相乘得出的结果作为半径 ε 确定 ε 近邻域。所有与 X_i 最近的 K_{base} 个样本点都被看作 X_i 的近邻，而 ε 近邻域内与 X_i 的距离大于 X_i 与第 K_{base} 个近邻点的距离的样本点，则用来决定 K_{add}。下面计算包含 K_{add} 个附加近邻的重构误差 ε，定义为

$$\varepsilon(K)=\sum_{i=1}^{N}\left|X_i-\sum_{i=1}^{N}W_{ij}X_j\right|^2$$

因为 K_{add} 对于不同样本点是变化的，所以 ε 可以看作 K 的函数。使重构函数取得最小值的 K 就是 LLE 算法可以选取的最优近邻个数。根据 Han 等人的研究成果，K_{base} 的最佳值为 7，常数 h 的最佳值为 1.2。

在改进近邻个数 K 选取的基础上，把经典 Isomap 流形学习算法中的距离定义引用到本文提出的算法中。新的距离定义方式可以使得处于样本分布较密集区域的样本之间的距离增大，而使得样本分布较稀疏的区域的样本之间的距离减

小，从而使样本集的整体分布均匀化，可以进一步降低 K 值的选取对实验结果的影响。改进后的基于新型近邻个数 K 的 LLE 算法描述如下：

Step1　对于高维空间中的每个样本点 X_i，首先给定 K_{base}（一般取 $K_{base}=7$）个近邻，计算 X_i 与其第 K_{base} 个近邻点的距离，并将结果与 h（一般取 $h=1.2$）相乘，结果作为 ε 邻域半径参数。将此区域中除 K_{base} 个近邻以外的样本点作为 K_{add} 选取的参考，根据重构误差函数式，选取使其最小的 K 值作为近邻个数。所有距离都采用经典 Isomap 流形学习算法中的距离定义，即

$$d_{ij}=\frac{\| X_i-X_j \|}{\sqrt{M(i)M(j)}}$$

式中，$M(i)$ 和 $M(j)$ 分别表示 X_i 和 X_j 到其 K 个近邻之间距离的平均值。

Step2　计算每个点 X_i 到其近邻点之间的权重 w_{ij}，即最小化：

$$\varepsilon(W)=\sum_{i=1}^{N}\| X_i-\sum_{j=1}^{N}w_{ij}X_j \|^2$$

式中，若 X_j 不是 X_i 的近邻，则 $w_{ij}=0$；$\sum_{j=1}^{N}w_{ij}=1$。

Step3　根据高维空间中的样本 X_i 和它的近邻 X_j 之间的权重 w_{ij} 来计算低维嵌入空间中的值。由于目标是在低维空间中尽量保持高维空间中的局部线性结构，而权重 w_{ij} 又代表局部信息，因此固定权重 w_{ij} 来使下面的损失函数最小化：

$$\phi(Y)=\sum_{i=1}^{N}\| Y_i-\sum_{j=1}^{N}w_{ij}Y_j \|^2=tr(Y^TMY)$$

式中，$\sum_{i=1}^{N}Y_i=0$ 且 $\frac{1}{N}\sum_{i=1}^{N}Y_iY_i^T=I$，$M=(I-W)^T(I-W)$。使上式最小化的解为由矩阵 M 的最小的几个特征值所对应的特征向量构成的矩阵。取 M 的最小的 $d+1$ 个特征值对应的特征向量，丢掉其中最小的特征值对应的特征向量，剩余的 d 个特征向量组成的矩阵就是得到的低维空间中的样本。

本文提出的近邻个数选取方法的优点如下：首先，与经典的 K 近邻算法相比，本文方法对于不同的样本点分别考虑它们的近邻个数，从而避免了样本点成簇，造成非连接区域和纠正错误的问题；其次，与经典的 ε 近邻算法相比，本文方法对不同的样本点设置不同的半径 ε，避免了不同样本点近邻个数相差过大的问题。

3　模拟实验及结果分析

为了说明本文提出的基于新型 K 近邻选取的 LLE 算法在云计算入侵检测背景下的优越性，下面利用 DARPA 提供的 KDDCUP99 入侵检测数据源[10]进行仿真实验。由于目前国际上对云计算的架构还没有统一的认识，同时云计算环境对计算能力和存储能力具有超高要求，故本文中只对所提模型的算法进行仿真实验。配置情况如下：CPU 为 Intel Pentium Dual E2140，内存 1G，使用 Matlab6.5 仿真实验环境，软件包为 Suykens 等提供的 LS-SVMlab1.5。

3.1　实验数据处理

DARPA 入侵检测数据集中用于训练的前 7 个星期的数据量过大，约为 500 万条记录，为了减少分类器 SVM 训练时间，必须从训练集中提取一小部分用于训练。为了使选择的 SVM 训练数据更具代表性，选取 MIT 实验室提供的 10percent 数据集共约 494021 条记录作为本节实验的训练数据。为了更好地进行仿真实验，验证本文提出的结合流形学习和支持向量机的分类算法的优越性，本文对数据进行了进一步处理：

(1)建立训练集。10percent 数据集共包括 494021 条记录，其中占 19.69%的约 97278 条记录被标记为 NORMAL，其余占 80.31%的约 396743 条记录被标记为入侵。入侵记录共包括 22 种攻击方式，并被划分为 DoS，PROBE，R2L，U2R 4 大类入侵方式。在实验中，将 97278 条 NORMAL 记录分别与 391458 条 DOS 记录、4107 条 PROBE 记录、1126 条 R2L 记录和 52 条 U2R 记录组合在一起，并按统一随机抽取的原则，借鉴 FangYie Leu 等人提取 2%的方法从 4 类组合中分别提取出约占 10percent 数据集 2%大小的 9880 条记录，从而建立了 4 个训练子集，分别命名为 DoS，PROBE，R2L，U2R 训练集。同时，利用相同的随机抽取方法从 10percent 数据集中直接抽取 9880 条记录，包括正常及各种入侵方式，将此训练子集命名为 ALL，具体训练集如表 1 所列。

表 1　训练集的组成

	NORMAL	DOS	PROBE	R2L	U2R	子集
DoS 训练子集	97278	391458	0	0	0	9880
PROBE 训练子集	97278	0	4107	0	0	9880
R2L 训练子集	97278	0	0	1126	0	9880
U2R 训练子集	97278	0	0	0	52	9880
ALL 训练子集	97278	391458	4107	1126	52	9880

(2)建立两类测试子集。将测试子集分为未包括未知攻击和包括未知攻击两组，其中第一组测试集只包括训练子集中出现的攻击方式，而第二组测试集除了包括训练集中所有的攻击方式之外，还加入了训练集中没有出现过的新型攻击方式。第二组测试集新加入的攻击方式如表 2 所列。

表 2　新型攻击方式列表

攻击类型	攻击方式
DoS	apache2，udpstorm，processtable，mailbomb
PROBE	snmpgetattack，named，xlock，xsnoop，sendmail，worm，snmpguess
R2L	xterm，ps，httptunnel，sqlattack
U2R	saint，mscan

由于在 10percent 数据集中 DoS 和 PROBE 攻击所占的比例已经比较大，而 R2L 和 U2R 攻击的比例太小，为了检验本文提出的方法对 R2L 和 U2R 的检测效果，在第二组测试集中增加了 R2L 和 U2R 类新型攻击的比重，具体比例如表 3 所列。

表 3　新型攻击方式比例列表

攻击类型	攻击样本总数	新型攻击样本数	新型攻击所占比重
DoS	229853	6555	2.9%
PROBE	4166	1789	42.9%
R2L	16189	10196	62.9%
U2R	228	189	82.8%

这样就建立了不含新型攻击的第一组测试集和含有新型攻击的第二组测试集，两组测试集的组成比例如表 4 所列。

表 4　测试集的组成

数据集	第一组测试集	第二组测试集
样本总数	489844	311029
正常样本数	97278	60593
异常样本数	392566	250436
DoS 样本数	388338	229853

PROBE 样本数	4115	4166
R2L 样本数	109	16189
U2R 样本数	4	228

3.2 改进模型与采用经典 LLE 预处理模型的实验比较

使用本文提出的改进 LLE 算法对 5 个训练子集进行降维处理并分别训练 5 个分类器，同时使用经典 LLE 算法对 5 个训练子集进行降维处理，并分别训练 5 个分类器。模型的平均训练时间如表 5 所列。

表 5 训练时间对比

	降维算法	ALL	DoS	PROBE	R2L	U2R
训练时间(s)	LLE	7	8.3	11	13	9
训练时间(s)	改进 LLE	28	41	89	24	77
检测时间(s)	LLE	3.6	1.2	2.9	4	1.8
检测时间(s)	改进 LLE	9	6	18	8	11

为了证明本文提出的采用改进的 LLE 流形学习算法的入侵检测模型比采用经典 LLE 算法的入侵检测模型具有更好的性能，利用包含新型攻击的测试集和未包含新型攻击的测试集分别对两类模型进行了实验比较。

在第一个实验中，将所有攻击都看作一种攻击，分别利用降维和未降维的 ALL 训练集训练，建立了两个入侵检测模型。对于每一个模型，分别使用两组测试子集，即只包含训练集中出现攻击的测试集和包含训练集中未出现攻击的测试集，进行了分类测试比较，得到的 ROC 曲线如图 3 所示。从图中可以看到，无论是对已知攻击还是新型攻击，利用改进算法建立的模型比利用经典算法建立的模型具有更高的 ROC 分值。

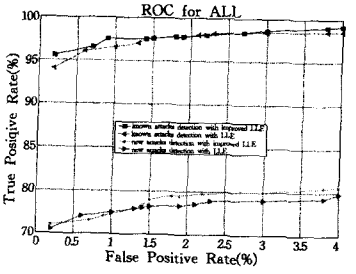

图 3 ROC for ALL

在第二个实验中，只考虑 DoS 类攻击，分别利用降维和未降维的 DoS 训练集建立了两个入侵检测模型。仍然用与第一个实验相同的测试集对两个模型分类器进行测试比较，得到的 ROC 曲线如图 4 所示。从图中可以看到，对于新型攻击的检测，利用改进算法建立的模型比利用经典算法建立的模型在检测准确率指标上高出近 5%。这表明利用改进算法建立的模型在检测新型攻击的准确率上具有优势。

在下面的几个实验中，仍然采用和第二个实验相同的实验方法，分别针对 PROBE，R2L，U2R 3 类攻击进行建模比较，得到的 ROC 曲线如图 5－图 7 所示。在图中可以看到，对于 PROBE，R2L，U2R 等攻击，本文提出的使用改进 LLE 算法的模型比使用经典算法的模型均具有更高的 ROC 值。这说明在同等条件下，本文提出的模型性能更加优越，当然这是以少量地增加训练时间为代价的。但是在云计算具有强大的计算能力的背景下，时间因素的约束会大大下降，所以以时间换取检测性能是值得的。

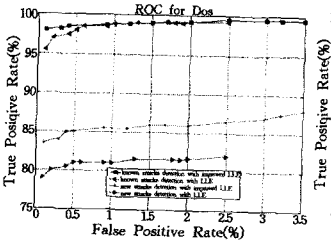

图 4 ROC for DoS

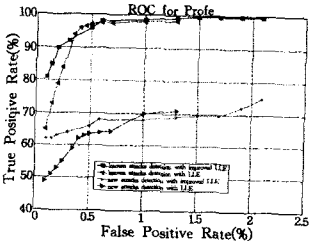

图 5 ROC for Probe

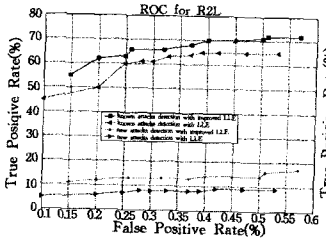

图 6 ROC for R2L

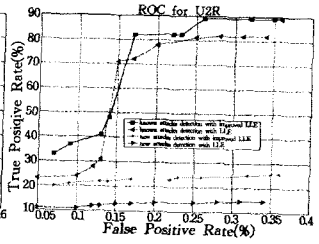

图 7 ROC for U2R

结束语 本文构建了适应云计算环境的分布式入侵检测系统框架，着眼于基于虚拟组织的入侵检测系统数据预处理模块设计，并将改进的非线性流形学习算法引入本课题提出的模型，作为特征提取模块对云计算环境下采集的网络数据进行预处理。仿真实验证明，本文提出的方法是可行和高效的。在未来的工作中，考虑对模型的其它模块进行详细设计和仿真，从理论和应用角度提高入侵检测模型的性能。

参 考 文 献

[1] 张健，曹蓟光. 互联网中云计算技术研究[J]. 电信网技术，2009，10(10)

[2] Jensen M, Schwenk J, Gruschka N, et al. On Technical Security Issues in Cloud Computing[C]// 2009. IEEE International Conference on Cloud Computing. Sept. 2009: 109-116

[3] 卿斯汉，等. 入侵检测技术研究综述[J]. 通信学报，2004，25(7)

[4] Leu Fang-Yie, Lin Jia-Chun, et al. A performance-based grid intrusion detection system[C]// 29th Annual International Computer Software and Applications Conference(COMPSAC 2005). Volume 1, July 2005: 525-530

[5] Schulter A, Vieira K, Westphall C, et al. Intrusion Detection for Computational Grids[C]// New Technologies, Mobility and Security(NTMS '08). Nov. 2008: 1-5

[6] Li Yuan-Cheng, Wang Zhong-Qiang. Wavelet Analysis and Pattern Recognition[C]// ICWAPR '07. Volume 4, Nov. 2007: 1462-1466

[7] Li Jun, Hao Pengwei. Hierarchical Structuring of Data on Manifolds[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR '07). June 2007: 1-8

[8] Foster I, Zhao Yong, Raicu I, et al. Cloud Computing and Grid Computing 360-Degree Compared[C]// Grid Computing Environments Workshop(GCE '08). Nov. 2008: 1-10

[9] Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding[J]. Science, 2000, 290: 2323-2326

[10] KDD Cup 1999 Data. Information and Computer Science[EB/OL]. http://kdd.ics.uci.edu/database/kddcup99/kddcup99.html