一种基于A_Kohonen算法的恶意代码自动分类机制

徐小龙[1,2] 熊婧夷[1] 王新珩[3] 王汝传[1]
(南京邮电大学计算机学院 南京 210003)[1] (南京大学计算机软件新技术国家重点实验室 南京 210046)[2]
(西苏格兰大学计算机学院 佩斯利 PA1 2BE)[3]

摘 要 目前海量的恶意代码报告已经成为基于云安全的反病毒网络系统的巨大负担。使用高效、科学的分类方法对大量涌现的已知或未知的恶意代码及其变种进行自动分类处理是快速应对恶意代码的基本前提。为了实现对恶意代码的自动分类,首先对解决聚类问题的经典无监督神经网络模型Kohonen算法进行改进,提出一种新的、引入部分监督学习过程的神经网络模型A_Kohonen算法;然后基于A_Kohonen算法实现对各种恶意代码的自动分类机制,从而为反病毒专家对恶意代码进一步细化与分析处理提供有效支持。实验分析表明,基于A_Kohonen算法的恶意代码自动分类机制能够有效、准确地初步分类恶意代码。

关键词 恶意代码,报告分类,反病毒,神经网络,信息安全
中图法分类号 TP309.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.08.039

Automatic Classification Mechanism of Malicious Code Based on A_Kohonen Algorithm

XU Xiao-long[1,2] XIONG Jing-yi[1] WANG Xin-heng[3] WANG Ru-chuan[1]
(College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)[1]
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China)[2]
(School of Computing, University of the West of Scotland, Paisley PA1 2BE, UK)[3]

Abstract The current mass of malicious code reports has become a huge burden of cloud-security-based anti-virus network systems. The utilization of efficient, scientific and automatic classification mechanism is the basic premise for responding quickly to large-scale known and unknown malicious codes and their new variants. In order to achieve the automatic classification of malicious codes, we improved the Kohonen algorithm, a classic neural network model with no mentors and proposed a new neural network model A_Kohonen with part supervised learning of the process. Then the A_Kohonen-based automatic classification mechanism of malicious codes was provided to support anti-virus experts to refine and analyze malicious codes further. Experimental analysis shows that the mechanism can initially classify malicious codes effectively and accurately.

Keywords Malicious code, Report classification, Anti-virus, Neural network, Information security

1 引言

恶意代码(Malicious codes)[1,2]是一组通过复制自身来感染其它软件的程序,包括传统的电脑病毒以及网络蠕虫、木马等。随着技术的发展,恶意代码的种类和数量均呈爆炸式发展的态势。传统反病毒软件系统对于层出不穷的恶意代码的反应存在着一定的滞后性[3]。为了弥补这一缺陷,尽快对互联网上出现的各类恶意代码作出及时反应,瑞星、趋势科技、卡巴斯基、McAFee、SYMANTEC、江民科技、PANDA、金山、360等都推出了各自的云安全(Cloud Security)[4,5]解决方案,通过网状的大量客户端对网络中软硬件行为的异常监测,获取恶意代码的最新制造、传播与感染信息,并传送到服务器端进行自动分析和处理,再快速把解决方案分发到每一个客户端。

而云安全系统成功实施与运行的先决条件显然是对海量用户提供的大规模恶意代码报告的分类、分析与汇总。例如趋势云安全系统[6]每天收集用户提交的2.5亿个恶意代码报告;卡巴斯基全功能安全软件以用户"知情并同意(Awareness & Approval)"的方式每日在线收集、分析数以万计的用户计算机提交的可疑报告;瑞星云安全的核心瑞星卡卡6.0每天

到稿日期:2013-10-02 返修日期:2013-12-16 本文受国家自然科学基金资助项目(61202004,61100199),江苏省自然科学基金项目(BK2011754),中国博士后科学基金资助项目(2012T50514,2011M500095,2013T60553),江苏省博士后科研资助计划项目(1102103C),江苏省高校自然科学研究计划资助项目(12KJB520007,13KJB520017),江苏省科技支撑计划(BE2013666),南京大学计算机软件新技术国家重点实验室资助项目(KFKT2012B21)资助。

徐小龙(1977-),男,博士,副教授,主要研究方向为计算机软件、分布式计算、信息安全、Agent技术等,E-mail:xuxl@njupt.edu.cn;熊婧夷(1988-),女,硕士生,主要研究方向为基于网络的计算机软件和信息安全技术等;王新珩(1968-),男,博士,教授,主要研究方向为网络计算、无线传感网、无线Mesh网络技术等;王汝传(1943-),男,教授,CCF会员,主要研究方向为计算机软件、网络计算和信息安全技术等。

可收集到 8～10 万个木马报告，然后对恶意代码进行分类和特征提取。

如此大规模恶意代码报告的分析对于反恶意代码系统来说是一个巨大的负担。不同的恶意代码因其生存平台、传播方式、潜伏周期、自身使命的不同而千差万别。要提高问题的解决效率，就要在反恶意代码的各个环节缩短处理代码的时间。使用高效、科学的自动分类方法对大量涌现的未知恶意代码和已知恶意代码新变种进行处理是快速应对恶意代码十分必要的基本前提。

本文的贡献在于：首先对解决聚类问题的经典无监督神经网络模型 Kohonen 算法进行改进，提出一种新的、引入部分监督学习过程的神经网络模型 A_Kohonen 算法；然后基于 A_Kohonen 算法实现对各种恶意代码的自动分类，为反病毒专家对恶意代码进行进一步细化与分析处理提供有效支持，从而提高解决恶意代码的效率。

2 改进的神经网络模型 A_Kohonen 算法

2.1 Kohonen 算法原理

Kohonen 算法[7-11]由芬兰学者 Teuvo Kohonen 提出，是一种基于自组织特征映射模型(Self-Organizing feature Map)的聚类网络，经常用于聚类分析。作为神经网络的分支，它通过自组织特征映射调整网络权值，使网络最终得以收敛于一种表示形态。在某一种形态中，一个神经元只对某种输入模式特别匹配或特别敏感，即特定的神经元可以成为某一输入模式的检测器。经过一定强度的网络训练，神经元将被划分为多个不同区域，不同区域对输入模型具有不同的响应特征。

Kohonen 聚类神经网络的工作原理为：在学习过程中，寻找最优匹配神经元，即对于竞争层上的神经元，计算其与输入的样本数据之间的欧几里德距离，然后进行相似性判断，算出最小欧几里德距离的神经元就是获胜的神经元；并相应修改获胜神经元本身及其邻域内其它神经元的权重，较邻近的神经元之间相互激励，而较远的神经元之间则互相排斥；最后竞争层各神经元的连接权系数经过自适应调整后，形成一定的分布，把数据之间的相似性表现在各类的神经元上，同类神经元具有相近的权系数，异类神经元之间的权系数则差异较大。在测试工作阶段，当输入新的样本数据时，计算该样本数据与各竞争层神经元之间的欧几里德距离，选择其中欧几里德距离最小的神经元作为输出结果，并输出该获胜神经元所属的类。

2.2 A_Kohonen 算法

Kohonen 神经网络是一种典型的双层前馈神经网络，它的模型可以在一维或二维的处理单元阵列上形成输入信号的分布拓扑图：第一层为输入层(又称匹配层)，该层的神经元个数要与样本向量位数一致，是单层单维度的神经元，计算输入模式向量与权向量之间的距离，即匹配程度；第二层为竞争层，该层的节点呈二维阵列分布，各神经元以匹配程度为依据进行竞争，确定匹配程度大(距离小)的神经元获胜。输入节点和输出节点以可变权值连接，如图 1 所示。

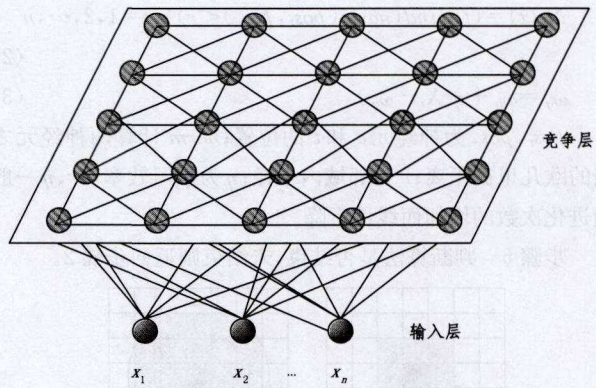

图 1 Kohonen 网络结构

Kohonen 算法是无监督的，虽然可以对未知类别数据进行无监督分类，但是分类结果中同一类别数据对应不同的网络节点，如果按照一个节点对应一类来说，Kohonen 网络分类的类别将比实际数据的类别多。为了改善分类效果，本文提出的改进的 A_Kohonen 算法在经过第一阶段的无监督学习后，加入一个有监督的学习过程，从而使系统工作向正确的分类结果调整。A_Kohonen 网络将竞争层与输出层分离，输出层结点个数同数据类别数目相同，每个节点代表一类数据。输出层节点和竞争层节点通过权值全连接，如图 2 所示。

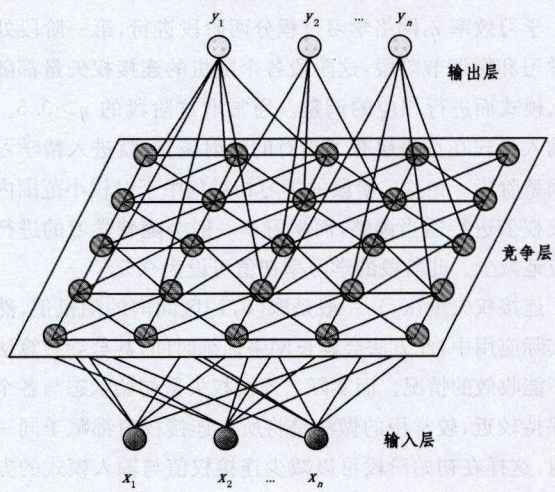

图 2 A_Kohonen 网络结构

算法的网络训练的步骤为：

步骤 1 网络初始化。初始化网络权值 ω_{ij}，其取值在[0，1]之间随机生成，$i=1,2,\cdots,N$；$j=1,2,\cdots,M$。确定 $\eta(t)$即学习效率的初始值 $\eta(0)(0<\eta(0)<1)$；确定邻域 $N_c(t)$的初始值 $N_c(0)$。邻域是指以获胜神经元为中心，同时包含几个神经元的一个区域范围，邻域[12]范围一般都是均匀对称的。常见的邻域如图 3 所示。

步骤 2 距离计算。计算输入向量 $X=(x_1,x_2,\cdots,x_n)$与竞争层神经元 j 之间的距离 d_j：

$$d_j=\left|\sum_{i=1}^{m}(x_i-\omega_{ij})^2\right|,j=1,2,\cdots,n \tag{1}$$

步骤 3 神经元选择。选择与输入向量 X 距离最小的竞争层神经元 c 作为获胜神经元。

步骤 4 权值调整。调整节点 c 与其邻域 $N_c(t)$内包含的节点权值：

$$N_c(t)=(t|find(norm(pos_t,pos_c)<r)),t=1,2,\cdots,n \quad (2)$$

$$\omega_{ij}=\omega_{ij}+\eta(X_i-\omega_{ij}) \quad (3)$$

pos_t，pos_c 为神经元 c 和 t 的位置；$norm$ 计算两神经元之间的欧几里德距离；r 为邻域内半径；η 为学习效率。r、η 一般随进化次数的增加而线性下降。

步骤 5　判断算法是否结束，未结束则返回步骤 2。

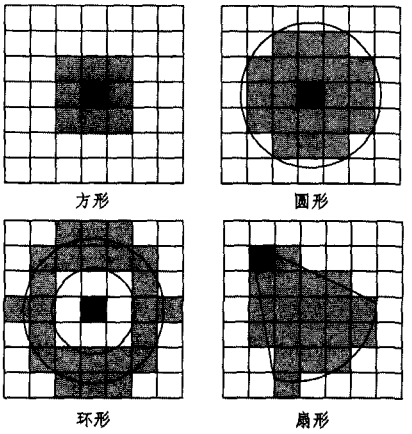

图 3　常见邻域示意图

在学习过程中，具体参数的控制如下[13-16]：

学习效率 η：网络学习过程分两阶段进行，第一阶段处于微学习和微调节阶段，这阶段各个随机的连接权矢量都随着输入模式而进行相应的调整。通常取这阶段的 $\eta>0.5$。一旦输入模式在竞争层有了相对的映射位置，就进入精学习和细调整阶段。在这一阶段的学习相对集中于对较小范围内的连接权值进行适当调整，而学习率一般会随着学习的进行而相应地减少。此阶段的学习率初值可设为 0.5。

连接权矢量 $\{\omega_{ij}\}$：一般是取[0,1]区间内的随机值，然而在实际应用中，此方法会延长网络训练时间，甚至导致算法出现不能收敛的情况。而实际上连接权矢量初始状态与各个模式保持较近，较常用的做法是将所有连接权值都赋予同一初始值，这样在初始阶段可以减少连接权值与输入模式的方向偏差。

邻域 $N_c(t)$：邻域规定了能够与获胜神经元 N 同时进行连接权调整的神经元范围。在初始学习阶段，邻域 $N_c(t)$ 所包括的范围较大，甚至可以覆盖整个输出层。随着学习的进一步进行，$N_c(t)$ 的范围将逐渐缩小，达到预期值。

数据输入 A_Kohonen 网络，在权值调整时，不仅调整输入层同竞争层优胜节点邻域内节点权值，而且调整竞争层优胜节点邻域内节点同输出层节点权值，调整公式为：

$$\omega_{jk}=\omega_{jk}+\eta_2(Y_k+\omega_{jk}) \quad (4)$$

其中，η_2 为二次学习因子；Y_k 为样本所属类别；ω_{jk} 为竞争层和输出层权值。这是 A_Kohonen 网络与传统 Kohonen 网络的训练过程的重要区别之一。

A_Kohonen 算法的流程图如图 4 所示。

经过一定强度的训练后，分类时首先计算出同未知样本最近的竞争层节点并将其作为获胜节点，与获胜节点连接权值最大的输出层节点代表类别就是最终输出的样本类别。

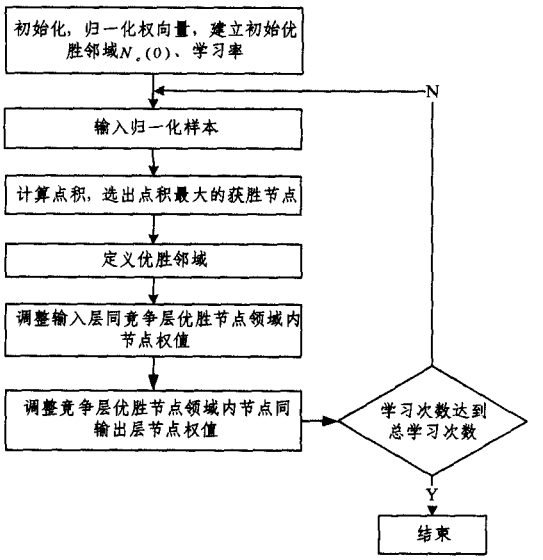

图 4　A_Kohonen 神经网络算法流程图

3　基于 A_Kohonen 的恶意代码自动分类

3.1　数据初始化

按照典型的恶意代码的分类和特征，本文将恶意代码归一化整理成 5 类和 18 个特征，如表 1 所列。

表 1　恶意代码数据类表

表项	类型	值
传播方式	数值型	自动传播 1，邮件传播 2，捆绑下载 3，黑客植入，多种方式共同作用(标记为 0)
降低安全级别	布尔型	是/否
自动发送邮件	布尔型	是/否
弹出广告网页或对话框	布尔型	是/否
下载恶意代码	布尔型	是/否
创建文件	布尔型	是/否
删除文件	布尔型	是/否
修改文件	布尔型	是/否
修改注册表	布尔型	是/否
修改浏览器主页	布尔型	是/否
隐私信息丢失	布尔型	是/否
系统信息暴露	布尔型	是/否
伪装图标	布尔型	是/否
连接指定网址	布尔型	是/否
禁用正常功能	布尔型	是/否
创建进程	布尔型	是/否
获取特殊权限	布尔型	是/否
监控摄像头	布尔型	是/否

用户在怀疑自己的主机遭受恶意代码侵害时，可以提交恶意代码报告。用户提交的报告表如表 2 所列，包括了大部分恶意代码攻击的可能症状。

数据归一化是指对样本数据进行归一化处理。网络初始化根据恶意代码数据特点初始化网络，样本中前 18 列为恶意代码特征，后 1 列为恶意代码类型。竞争层分类节点代表输入数据潜在的分类类别，一般都要远远大于实际类别，选择竞争层的节点数为 16 个，排列在一个 4 行 4 列的矩阵中。以下是数据预处理的相关代码：

```
%数据归一化
inputn=input;
```

```
[nn,mm]=size(inputn);
[b,c]=sort(rand(1,nn));
%网络期望输出
for i=1:nn
    switch attackkind(i)
        case 1
            output(i,:)=[1 0 0 0 0];
        case 2
            output(i,:)=[0 1 0 0 0];
        case 3
            output(i,:)=[0 0 1 0 0];
        case 4
            output(i,:)=[0 0 0 1 0];
        case 5
            output(i,:)=[0 0 0 0 1];
    end
end
```

表 2　用户提交报告表

表项	类型	值
碰到恶意报告，系统、防火墙或反病毒程序未发出警告	布尔型	是/否
邮箱软件自动向外发送邮件	布尔型	是/否
系统弹出广告窗口、广告网页或不明对话框	布尔型	是/否
磁盘出现恶意程序	布尔型	是/否
不明文件增多	布尔型	是/否
正常文件消失或被隐藏	布尔型	是/否
文件无法正常打开或运行	布尔型	是/否
注册表发现不明项	布尔型	是/否
账号丢失	布尔型	是/否
系统信息暴露	布尔型	是/否
点击常用图标无法正常响应	布尔型	是/否
网络自动访问某 IP	布尔型	是/否
系统或软件功能无法正常使用	布尔型	是/否
出现不明进程	布尔型	是/否
机器被不明主机远程访问	布尔型	是/否
摄像头偶尔自动打开	布尔型	是/否
安全模式下蓝屏	布尔型	是/否
网速变慢	布尔型	是/否

3.2　训练与测试

训练时按照式(1)选择出与输入样本最接近的竞争层节点作为该样本的获胜节点。再按照式(2)调整获胜节点半径 r 内节点权值，其中邻域半径和学习速率随着进化过程逐渐变小，这样输入数据逐渐向几个节点靠近，使网络具有聚类的功能。而竞争层汇聚出几个结果后，再由式(4)向输出层汇聚。网络在学习后便拥有了对恶意代码样本的分类能力。其中训练的数据是从恶意代码样本数据中随机抽取的。网络训练完成后，随机抽取数据看网络是否已经具有对恶意代码样本的分类能力。相关代码如下：

```
%% 迭代求解
maxgen=10000;
for i=1:maxgen
%自适应学习率和相应半径
ratel=ratelmax-i/maxgen*(ratelmax-ratelmin);
rate2=rate2min+i/maxgen*(rate2max-rate2min);
r=rlmax-i/maxgen*(rlmax-rlmin);
%从数据中随机抽取
k=unidrnd(100);
x=input_train(k,:);
y=output_train(k,:);
%计算最优节点
[mindist,index]=min(dist(x,wl));
%计算周围节点
dl=ceil(index/4);
d2=mod(index,4);
nodeindex=find(dist([dl d2],jdpx')<=r);
%权值更新
for j=1:length(nodeindex)
    wl(:,nodeindex(j))=wl(:,nodeindex(j))+rate1*(x'-wl
    (:,nodeindex(j)));
    w2(nodeindex(j),:)=w2(nodeindex(j),:)+rate2*(y-w2
    (nodeindex(j),:));
    end
end
%% 聚类结果
Index=[];
for i=1:100
    [mindist,index]=min(dist(inputn(i,:),wl));
    Index=[Index,index];
end
  inputn_test=kohonen(:,1:18);
%样本验证
for i=1:650
    x=inputn_test(i,:);
    %计算最小距离节点
    [mindist,index]=min(dist(x,wl));
    [a,b]=max(w2(index,:));
    outputfore(i)=b;
end
```

4　实验分析

本文从 www. viruschina. com 选择 650 个恶意代码样本：其中 1－100 属于黑客病毒(类别标签为 1)，101－200 属于宏病毒(类别标签为 2)，201－300 属于脚本病毒(类别标签为 3)，301－400 属于木马(类别标签为 4)，401－650 属于蠕虫(类别标签为 5)。训练数据和测试数据都存储在文件 kohonen. mat 中，其中前 18 列为恶意代码特征，第 19 列为恶意代码种类。

本文将最大邻域半径 rlmax 设为 1. 3，最小邻域半径 rlmin 设为 0. 02，由输入层向竞争层进化的最大学习率 ratelmax 设为 0. 07，最小学习率 ratelmin 设为 0. 02；由进化层向输出层净化的最大学习率 rate2max 设为 0. 3，最小学习率 rate2min 设为 0. 1。网络总共学习 10000 次。

为了进行对比分析，分别用训练数据训练无监督的 Kohonen 网络和本文提出的 A_Kohonen 网络。网络在学习后便拥有了对未知样本的分类能力。网络训练完成后，随机抽取数据看网络是否已经具有对恶意代码样本的分类能力。

将 650 个恶意代码样本数据输入到无监督的 Kohonen

网络中，得到如图5所示的结果，可以看到各类恶意代码会各自汇聚到一个特定值。

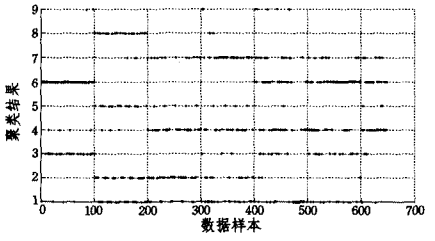

图5 Kohonen网络分类结果

因为无监督神经网络本身的属性限制，所以采用传统的Kohonen网络并不能将数据分成定义的某几类，不符合恶意代码样本分类的实际应用。通过建立带有监督性质的A_Kohonen网络，使用训练过的A_Kohonen网络对样本数据进行分类，分类仿真实验效果如图6所示。

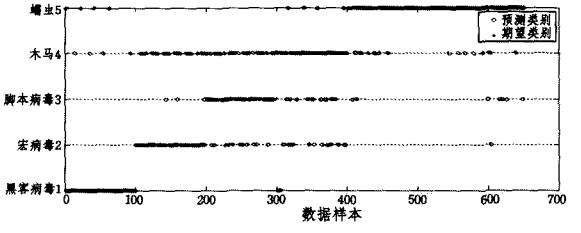

图6 基于A_Kohonen网络的恶意代码分类示意图

如图6所示，实心点表示恶意代码的期望类别即实际类别，空心圆表示经过A_Kohonen网络分类后输出的类别，当它们两两重合时，说明分类成功。使用A_Kohonen可以将集中恶意代码进行粗略的分类，其中对蠕虫和黑客病毒的分类较为准确，一是因为相较于其它恶意代码来说，蠕虫和黑客病毒特征比较明显，比如蠕虫必须具有主动复制并传播的特性，而黑客病毒一般都要获取系统权限。而木马和脚本病毒的特征比较分散，因为在真实的网络系统中，本来就存在可以被称为木马的脚本病毒，所以可能经过系统分类出来的结果并不完全符合预期，这也是符合实际情况的。

而采用神经网络进行恶意代码分类，本来就是为了将恶意代码进行一个初期分类，以便于专家进行系统化处理，对于界限不分明的恶意代码，分在任一领域实际上并不会影响实际工作进度。而在实际应用中，可以将恶意代码多次输入已训练好的网络，取大概率的预期类别对其进行分类。

结束语 以恶意代码行为划分恶意代码，并不是说以恶意代码特征来诊断恶意代码，而主要是用于对用户提交的恶意代码特征报告做一个初级的分类。本文基于改进的A_Kohonen算法，实现了基于恶意代码特征的初步分类，细化了恶意代码处理工作的责任分配，方便专家对恶意代码作进一步逐类处理。本文的仿真实验包含了5种典型的恶意代码样本数据，有效地证明了本文提出的分类机制是确实可行的。在实际应用中，对于神经网络本身而言，需要大量数据的输入才能更有利于神经网络在培训后模拟出更贴切的输出结果。

参考文献

[1] 梁晓. 恶意代码行为自动化分析的研究与实现[D]. 成都：电子科技大学，2008

[2] 杨婷. 基于行为分析的恶意代码检测技术研究与实现[D]. 成都：电子科技大学，2010

[3] Xu X L, Wang R C, Xiao F. Malicious code passive propagation model and vaccine distribution model of P2P networks[J]. Journal of Systems Engineering and Electronics, 2010, 21(1): 161-167

[4] 百度百科. 杀毒软件[EB/OL]. http://baike.baidu.com/view/33433.htm, 2012-08-12

[5] 互动百科. 云安全[EB/OL]. http:/ www.hudong.com/wiki/云安全, 2011-07-28

[6] Trend Micro. AntiVirus[EB/OL]. http://cn.trendmicro.com, 2011-01-26

[7] Stavrou E, Spiliotis S, Charalambous C. Flexible working arrangements in context: An empirical investigation through self-organizing maps[J]. European Journal of Operational Research, 2010, 202(3): 893-902

[8] Kokhanenko I K. Fractal-clustering analysis of video information [J]. Journal of optical technology, 2010, 77(8): 499-503

[9] Fernandez-Varela R, Andrade J M, Muniategui S, et al. Identification of petroleum hydrocarbons using a reduced number of PAHs selected by Procrustes rotation[J]. Marine Pollution Bulletin, 2010, 60(4): 526

[10] Carrieri A H, Copper J, Owens D J, et al. Infrared differential-absorption Mueller matrix spectroscopy and neural network-based data fusion for biological aerosol standoff detection[J]. Applied Optics, 2010, 49(3): 382-393

[11] Onuki Y, Takayama K. Phase behavior in a ternary lipid membrane estimated using a nonlinear response surface method and Kohonen's self-organizing map[J]. Journal of Colloid and Interface Science, 2010, 343(2): 628-633

[12] 董琦. 基于表观模型和邻域信息的图像区域分类与识别算法[D]. 北京：北京大学，2009

[13] 黎洪松. 一种新的自组织神经网络算法[J]. 北京师范大学学报：自然科学版，2005，41(5)：496-498

[14] 郑明文. 径向基神经网络训练算法及其性能研究[D]. 青岛：中国石油大学(华东)，2009

[15] 段隆振，朱敏，王靓明. 基于双Kohonen神经网络的Web用户访问模式挖掘算法[J]. 计算机工程与科学，2009，31(9)：95-98

[16] 朱敏，段隆振，王靓明. 一种基于Kohonen神经网络Web用户行为模式的挖掘方法[J]. 南昌大学学报：理科版，2009，33(6)：591-594