基于 Elman 网络的未知恶意代码检测系统研究与实现*)

Study and Implementation on Detecting System for Unknown Malicious Code Based on Elman Network

张波云[1,2]　殷建平[1]　唐文胜[1]　嵩敬波[1]

(国防科技大学计算机学院　长沙 410073)[1]　(湖南公安高等专科学校计算机系　长沙 410138)[2]

Abstract　We design an intelligent detecting system to recognize unknown computer virus. Our method is based on Elman network classification algorithm. Using this method, a malicious executable code detection network model is designed also. The model can detect known and unknown malicious code by analyzing their behavior. We gather 423 benign and 209 malicious executable programs that are in the Windows PE format, and trace the API calls of each program. After selecting the most relevant API calls as feature, we evaluate the Elman network classification algorithm to detecting computer virus.

Keywords　Malicious code, Elman neural network, API function, Virus detection

1　引言

当前的计算机病毒检测技术主要基于特征检测法，该法只能用于检测已知的病毒，对于新出现的病毒的检测无能为力[1]。长期以来，为解决这一问题各反病毒研究机构都在努力探讨病毒检测的智能方法。

基于神经网络的病毒检测系统技术如今已经在国内外的一些大型的杀毒防毒软件中得到应用。如1998年 Symantec 公司宣布把 IBM 公司专利的神经网络引导检测技术[2]集成到诺顿防病毒(Norton AntiVirus)产品中。M. Schultz 等[3]曾提出用数据挖掘的算法如朴素贝叶斯算法等检测未知恶意代码。

基于上述思想的启发，本文提出一种基于 Elman 神经网络[4]分类算法的检测方法来实现对计算机病毒的近似判别。

2　模型结构

病毒检测系统结构框图如图1所示。系统由病毒防火墙、应用服务器和病毒检测服务器组成，进入系统的可疑文件首先经过系统的第一道防线病毒防火墙的检测，若没有检测出病毒：则将其复制两份，一份拷贝直接存放于应用服务器中；一份拷贝则进入检测服务器中，对其行为特征进行检测，如果发现其可疑行为，判断为感染了未知病毒，检测服务器则将信息反馈给应用服务器，然后在应用服务器中对该文件进行隔离、监管或删除。

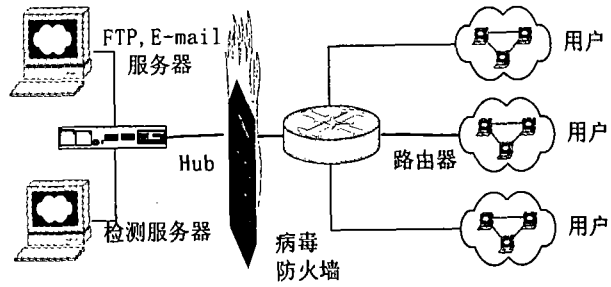

图1　系统结构框图

检测服务器中的病毒检测引擎基于 Elman 网络实现，其分类算法主要针对程序的执行过程中使用的系统 API 函数的调用，它监视程序的行为并进行分析，能有效地检测未知病毒和各种多态与变形病毒。检测服务器自身的安全防护，可以采用系统还原保护法等实现。

3　用于识别恶意代码的 Elman 神经网络

Elman 网络是一种回归神经网络，结构如图2所示。在 Elman 网络中回归层采用 tansig 神经元，输出层采用线性神经元，这样 Elman 神经网络就可以按照任何精度逼近任何函数，并可以采用 BP 算法进行训练。因为网络结构中初始权值的选取、隐含层节点数的确定以及权值的调整算法对 BP 算法学习速度和收敛性都有很大的影响，而且各参数的调节没有一定的规律可循，下面就 Elman 网络在恶意代码识别中的应用，对这些问题分别进行讨论。

*)基金项目：国家自然科学基金第 60373023 号资助。张波云　博士生，研究方向为信息安全；殷建平　教授，博士生导师，研究方向为算法设计与分析、模式识别与人工智能、信息安全等；唐文胜　博士生，研究方向为信息安全；嵩敬波　博士生，研究方向为信息安全。

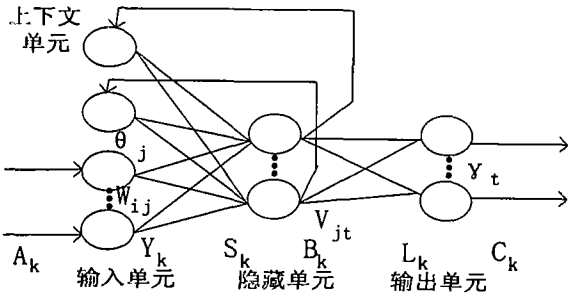

图 2 Elman 网络结构

图中输入模式向量 $A_k=(a_1,a_2,\cdots,a_n)$，希望输出向量 $Y_k=(y_1,y_2,\cdots,y_q)$，中间层输入向量 $S_k=(s_1,s_2,\cdots,s_p)$，输出向量 $B_k=(b_1,b_2,\cdots,b_p)$，输出层单元输入向量 $L_k=(l_1,l_2,\cdots,l_q)$，输出向量 $C_k=(c_1,c_2,\cdots,c_q)$，输入层至中间层连接权$\{W_{ij}\}$，i=1，2，…，n. j=1，2，…，p；中间层至输出层连接权$\{V_{jt}\}$，j=1，2，…，p. t=1，2，…，q，中间层各单元输出阈值$\{\theta_j\}$，j=1，2，…，p，输出层各单元输出阈值$\{\gamma_t\}$，t=1，2，…，q，以上 k=1，2，…，m。

3.1 初始权值的选取

Elman 网络的初始权值一般选取[－1，1]之间的随机数，但通过大量的实验研究，发现初始权值的选择不仅和网络的结构有关而且和节点输入的取值范围有关。

由于网络的结构中隐含层节点和输出层节点的输出范围为 0-1 之间的浮点数，所以为了有利于总结初始权值选取的规律，在这里对神经网络输入的节点的取值范围作归一化处理，使输入节点的取值范围也为 0-1 的范围。根据我们的实验一般使网络训练的初始权值取之间的随机数，其中 I 为该连接权值的输入节点的数目。

3.2 权值的学习算法

为防止学习过程出现振荡这种情况，我们采用学习步长自适应的方法来调节 BP 网的连接权值，在标准的 BP 的学习算法中加入一条学习步长的自适应算法，改善学习收敛效果。

开始学习步长选择一般取 $1/(10I_{max})$，其中 I_{max} 为输入节点最大幅值。令 $D(k)=-\frac{\partial E}{\partial W(k)}$ 表示 k 时刻的负梯度，η 为学习率，$\eta>0$。令 S=D(k)D(k－1)，权值和学习步长的调节规则如下：

$w(k+1)=w(k)+\eta D(k)$；$\eta(k+1)=2^{\lambda}\eta(k)$；当 $s>s_1$ 时 $\lambda=1$；当 $0<s<s_1$ 时 $\lambda=0$；当 $s\leqslant 0$ 时 $\lambda=-1$ 且 $w(k+1)=w(k)+\alpha w(k)$；其中 s_1 是一取值为 0.1 左右的正数，α 为±0.05 之间的随机数。

3.3 隐含层数及其节点数的选择

对于一神经网络，当训练样本确定之后，输入层和输出层节点数自然随之而定，但隐含层节点数目的选取却十分盲目。

我们通过在恶意代码检测系统中对 Elman 网络的大量训练，认为隐含层节点的数目不但和网络输入输出节点的数目有关，而且和学习样本的数量有关，经过大量实验验证隐含节点数的大约取值可根据以下经验公式确定 $H=\sqrt{I+O+S/O}$ 其中 H、I、O、S 分别表示隐含层、输入层、输出层节点数和训练样本总数。

4 未知恶意代码检测

4.1 病毒特征提取

病毒与一般程序的区别是要在于执行了一些特殊的动作来破坏系统。它们都是一种程序，需要调用操作系统提供的各种功能函数才能达到传播自身和破坏系统的目的。因此我们将程序在所使用的 DLL 中调用的 API 函数做为待检测程序的行为特征。

对样本库中的程序进行 API 调用跟踪处理后，可以得到大量的系统调用序列，这些不同的 API 对于病毒识别所起的作用是不一样的。在此，我们采用概率统计学中的均方差法来进行实验。它能较好地体现各不同的 API 调用的“贡献程度”，一个被调用 API 的分类作用与它的类间频率均方差成正比。其计算过程如下：(1)对所有训练库的样本进行系统调用跟踪，获得调用的 API 序列 $A=\{A_1,A_2,\cdots,A_i\}$，$(1\leqslant i\leqslant p)$统计各 API(即 A_i)在每一病毒程序 V_j 中出现的频率 A_{ij}^{v}及在每一正常程序 N_j 中出现的频率 A_{ij}^{N}；(2)计算每个被调用的 API 在病毒程序与正常程序文件中出现的频率均值 $E(A_i^v)=\frac{1}{S}\sum_{j=1}^{s}A_{ij}^{v}$，s 为病毒样本数量，$E(A_i^N)=\frac{1}{n}\sum_{j=1}^{n}A_{ij}^{v}$，n 为正常程序数量；(3)计算每个 API 的总出现频率均值；$E(A_i)=\frac{E(A_i^v)+E(A_i^N)}{2}$；(4)计算每个 API 的类间频率均方差 $D(A_i)=\sqrt{(E(A_i)-E(A_i^v))^2+(E(A_i)-E(A_i^N))^2}$。

然后，将所有的 API 按均方差大小排序，选出前 t 个组成特征。我们通过实验后选取了 88 个主要 API，一个示例如表 1 所示。输入 Elman 网络的

表 1 特征列表示例

序号	程序行为	相关 API 调用	危险度	动态链接库
1	写注册表	RegSetValueExA RegCreateKeyExA RegDeleteKeyA	☆☆	KERNEL32. dll
2	文件搜索	FindClose FindFirstFileA FindNextFileA	☆☆☆	KERNEL32. dll

数据是各 API 函数调用的频率。在实验中，我们利用标准化方法对这些原始数据先进行归一化预处理。

4.2 病毒检测引擎

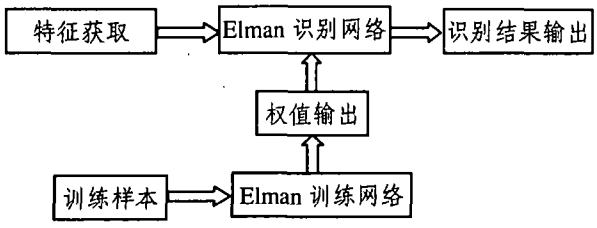

图 3 Elman 网络病毒识别算法框图

基于 Elman 网络的病毒识别过程如图 3 所示。系统首先对网络进行训练，详细过程如下：

(1)对训练库的全部样本获取其 API 调用序列；(2)计算每个 API 函数在病毒程序集中出现的频率以及在正常程序集中出现的频率，然后求得类间频率均方差；(3) 根据类间频率均方差值降序排列，选取前 t 个 API，构成特征向量；(4)对各特征值进行归一化处理；(5)设置 Elman 网络初始参数，将数据输入网络，根据各样本标记及期望误差 ε 对网络进行训练；(6)输出权值，训练结束。

然后，根据训练过程获得的参数，修正 Elman 识别网络的初始值，对待检测样本进行分类，具体过程如下：(1)获取待测程序的 API 调用序列；(2)计算各特征量出现的频率，并进行归一化处理；(3)将特征值输入 Elman 网络进行识别；(4)输出待检测程序类别，即是否病毒，算法结束。

5 实验分析

表 2 用于实验的样本数据

	样本空间	训练集	测试集
正常程序	423	373	50
染毒程序	209	159	50
合计	632	532	100

用于实验的样本数据如表 2 所示。正常程序选用的是 windows 2000 server 首次安装后，机器中的全部 PE 文件共 423 个。我们从以下网站获得测试病毒样本程序 209 个：http://vx.netlux.org，http://www.cs. Columbia. edu / ids/mef/。

为获得样本空间中程序的特征向量，我们编写了 API 调用跟踪器，可以实现 Window 2000 Server 环境下的全部 API 函数调用的拦截。实验目标主要是对查病毒引擎的错误率进行测试。我们将错误类型分为二种，即(1)将正常程序判断为病毒，称为 False Negative，(2)将病毒程序判断为正常程序，称为 False Positive。

实验中 Elman 网络的输入层神经元 88 个，输出层有 2 个神经元，隐藏层有 18 个神经元。以归一化后的病毒特征向量做为网络输入，以程序类别{病毒，非病毒}做为输出，第一层初始学习率 η 为 1.5，第二层初始学习率为 1.0，期望误差 ε 为 0.01。实验结果见表 3。

表 3 实验结果

样本数	学习时间 (h)	False Negative	False Positive
532	0.37	4.86%	4.24%
450	0.23	5.62%	5.27%
250	0.18	7.75%	7.41%
100	0.15	6.33%	8.35%

我们曾[5]利用基于实例学习的 K-最近邻算法对本组样本空间中的数据进行过测试，其检测错误率最小约为 4.8%。使用 Elman 神经网络分类算法得到效果稍好，但 KNN 算法开销很小，而神经网络算法的开销则大得多，实际应用中这二方面都要兼顾。

结论 本文尝试用基于 Elman 网络的分类算法来实现对计算机病毒的检测，在此基础上，设计了一个恶意代码检测网络模型，它结合了特征码检测技术与人工神经网络技术，检测成功率较高。在我们的测试床中，将计划进行其它的基于机器学习算法的恶意代码检测实验，并对各种方法的性能与开销进行对比选择和优化以期用于实际的产品中。

参 考 文 献

1 Spinellis D. Reliable Identification of Bounded-LengthViruses Is NP-Complete . IEEE Transactions on information Theory, 2003,49(1):280～284

2 Tesauro G J, Kephart J O , Sorkin G B. Neural networks for computer virus recognition . IEEE Expert,1996 (8) : 5～6

3 Schultz M, Eskin E, Zadok E, Stolfo S. Data mining methods for detection of new malicious executables. In Proceedings of the IEEE Sympo- sium on Security and Privacy, Los Alamitos, CA, IEEE press, 2001. 38～49

4 Haykin S. Neural networks : a compre- hensive foundation, New- York :Prentice Hall PTR,2004

5 张波云，殷建平，张鼎兴，蒿敬波. 基于 K-最近邻算法的未知病毒检测. 计算机工程与应用，2005,41(6):7～10