DOI:10.3969/j.issn.1673-4785.201210055
网络出版地址:http://www.cnki.net/kcms/detail/23.1538.TP.20130409.1436.002.html

基于神经网络的僵尸网络检测

蒋鸿玲,邵秀丽
(南开大学 信息技术科学学院,天津 300071)

摘 要:目前主流的僵尸网络检测方法主要利用网络流量分析技术,这往往需要数据包的内部信息,或者依赖于外部系统提供的信息或僵尸主机的恶意行为,并且大多数方法不能自动存储僵尸网络的流量特征,不具有联想记忆功能.为此提出了一种基于BP神经网络的僵尸网络检测方法,通过大量的僵尸网络和正常流量样本训练BP神经网络分类器,使其学会辨认僵尸网络的流量,自动记忆僵尸流量特征,从而有效检测出被感染的主机.该神经网络分类器以主机对为分析对象,提取2个主机间通信的流量特征,将主机对的特征向量作为输入,有效地区分出正常主机和僵尸主机.实验表明,该方法的检测率达到99%,误报率在1%以下,具有良好的性能.

关键词:僵尸网络;BP神经网络;特征向量;网络流量;检测算法
中图分类号:TP393 **文献标志码**:A **文章编号**:1673-4785(2013)02-0113-06

Botnet detection algorithm based on neural network

JIANG Hongling, SHAO Xiuli
(College of Information Technical Science, Nankai University, Tianjin 300071, China)

Abstract: The most current botnet detection algorithm are typically based on network traffic analyzing technologies that usually need packet payload. The botnet detection algorithm also relies on information obtained by external systems or malicious behaviors of bots that do not automatically store the features of botnet traffic and do not have the ability of associative memory. As a result, this paper proposes a botnet detection algorithm based on BP neural network which trains the BP neural network classifier through a lot of botnet and normal traffic samples and allows it to learn how to identify botnet traffic and automatically remember the features of botnet traffic and therefore, detect the infected hosts effectively. The neural network classifier takes the host-pairs as analysis objects, extracts the traffic features of communications between two hosts and takes the feature vectors of host-pairs as input, thus, effectively distinguishing the normal hosts and bots. The experimental results show that the detection rate of our algorithm can achieve to 99% and the false positive rate is below 1% and the algorithm has a good performance.

Keywords: botnet; BP neural network; feature vector; network traffic; detection algorithm

近年来,僵尸网络的快速发展使因特网面临严重的安全威胁.僵尸网络是攻击者(botmaster)通过传播僵尸程序控制大量主机,通过命令与控制信道(command and control,C&C)与僵尸主机通信并发布命令[1].攻击者利用僵尸网络可发起多种攻击,如分布式拒绝服务攻击、垃圾邮件、信息窃取等[2].

目前主流的僵尸网络检测方法是通过分析网络流量来检测.文献[3]通过昵称检测IRC僵尸网络;文献[4]通过PageRank算法计算主机级别,再根据已知的僵尸网络信息进行检测;文献[5]通过网络通信图识别P2P网络,再利用外部系统提供的信息区分合法P2P网络与P2P僵尸网络;文献[6-7]通过识别僵尸主机的恶意行为检测僵尸网络.这些僵尸网络检测方法或者需要数据包的内部信息,无法检测加密的僵尸网络;或者依赖外部系统提供信息,不能独立进行检测;或者依赖僵尸主机的恶意行为,

收稿日期:2012-10-26. 网络出版日期:2013-04-09.
基金项目:国家科技支撑计划基金资助项目(2012BAF12B00);天津市重点基金资助项目(11jczdjc28100).
通信作者:邵秀丽. E-mail:shaoxl@nankai.edu.cn.

在僵尸主机不发起攻击时,不能有效检测出僵尸网络.大多数现有的检测方法无法自动学习僵尸的网络流量特征,检测系统无法自动进行检测学习,没有联想记忆功能,因此本文提出基于神经网络的僵尸网络检测方法,用大量的样本训练神经网络分类器,然后用训练好的神经网络分类器模型检测僵尸网络.本文的检测方法的优点是:1)神经网络的输入数据只需要数据包头部信息,不需要数据包的内部信息,可以检测加密的僵尸网络;2)基于神经网络的检测系统不依赖外部系统提供僵尸网络的信息,能独立完成检测;3)神经网络分类器关注僵尸主机的控制命令流量,不依赖僵尸网络的恶意行为,在僵尸主机处于"发呆"状态下仍能检测出僵尸网络;4)训练好的神经网络分类器能自动记忆僵尸网络的流量特征,当新的流量达到,能快速得出检测结果.

1 僵尸网络流量的特征分析

要通过网络流量检测僵尸网络,首先要明确僵尸网络流量的特征.僵尸主机既有恶意攻击行为也有C&C通信[6].僵尸主机不会持续不断地进行恶意攻击,在Botmaster给僵尸主机发布命令前,僵尸主机保持"发呆"状态,不发起任何攻击[8].此外,僵尸主机的攻击往往很隐蔽,很难通过网络流量检测出来[9].相比之下,C&C通信存在于僵尸网络的整个生命周期中,因此本文通过识别僵尸网络的C&C流量进行检测.

为了有效识别出僵尸网络的C&C通信,首先确定最能区分出僵尸网络的C&C通信流量和正常流量的特征.被僵尸网络感染的用户并不知道自己已被感染,仍然进行正常的网络活动,同时僵尸程序偷偷在后台执行,因此僵尸主机既有正常的流量也有僵尸网络流量.如果以被测网络的主机作为分析对象,僵尸网络流量的特征会被大量正常流量覆盖,因此本文将被测网络内部主机和外部主机作为一个主机对,以主机对作为分析对象,这样可以有效分离出与内部主机通信的合法主机和僵尸主机.

为提取僵尸网络主机对的特征,首先分析僵尸网络的主机对和正常主机对的区别.僵尸主机为了获取命令或者更新,需要频繁/周期性地连接C&C服务器或者其他僵尸主机[10-11].与正常流量相比,僵尸主机间的C&C通信一般字节数较小,并且持续时间较短.而主机对的一次通信会形成一个流,因此主机对的通信特征可通过分析主机对间流的特征提取出来.由于僵尸网络通常采用TCP协议通信,因此本文只分析TCP流,并提取主机对的TCP流的特征,而不考虑其他僵尸网络流量特征,本文提取主机对的特征包括:

1)通信次数,即TCP流个数;

2)前后2次通信的时间间隔的平均值;

3)前后2次通信的时间间隔变化的平均值;

4)TCP流的字节数平均值;

5)TCP流的数据包个数平均值;

6)TCP流的持续时间平均值.

提取出这些特征后,每个主机对即可用1个六维的特征向量表示,特征向量的每一维表示主机对的一个特征.

2 基于神经网络的僵尸网络检测模型

首先用大量样本训练神经网络分类器,然后用训练好的神经网络分类器检测僵尸网络.图1为基于神经网络的僵尸网络检测模型.

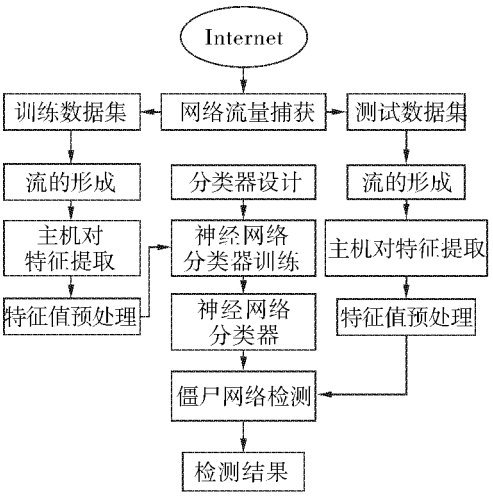


图1 基于神经网络的僵尸网络检测模型

Fig.1 Botnet detection model based on neural network

基于神经网络的僵尸检测过程如下:

1)捕获网络流量.主要获取被测网络内部主机和外部主机通信的数据包.2)形成流.2个主机间的一次通信会形成一个流,因此将原始数据包转换为流.3)提取主机对的特征.主机对是指2个互相通信的主机,即被测网络内部主机及与其通信的外部主机.本文的神经网络检测模型以主机对作为分析对象,是为了分离出与内部被感染主机通信的合法主机和僵尸主机.4)特征值预处理.对提取的主机对特征进行预处理,通过归一化操作使各个特征值的取值在同一范围内.5)分类器设计.根据特征向量设计神经网络分类器各层神经元的个数.6)神经网络分类器训练.利用具有类标签的训练数据集对神经网络进行训练,得到训练好的神经网络分类器.7)僵尸网络检测.利用训练好的神经网络分类器检

测测试数据集中的僵尸网络,得到检测结果.

其中,1) ~4)是数据准备及预处理;5) ~6)是神经网络分类器的形成;7)是利用训练好的分类器检测僵尸网络.

2.1　数据准备及预处理

为了得到检测性能好的神经网络分类器,首先要准备训练数据并进行预处理. 训练数据要能体现出僵尸网络流量的特征,区分出正常流量和僵尸网络流量. 训练数据准备及预处理包括 3 个过程,首先是网络流量捕获,然后是流的形成,最后是主机对特征的提取和预处理.

2.1.1　网络流量捕获

网络流量捕获主要抓取被测网络内部主机与外部主机通信的数据包. 图 2 为网络流量捕获图. 本文通过交换机端口镜像技术采集真实网络环境中的流量. 流量采集器运行 tcpdump 工具捕获被测网络内主机访问 Internet 的流量,由于僵尸网络通常采用 TCP 协议通信,因此只采集 TCP 协议的数据包,每个小时新建一个 pcap 文件存储数据包. 数据解析器将采集的原始数据解析为文本文件,导入数据库中. 真实网络环境中的流量都标记为非僵尸网络流量,僵尸网络流量通过模拟产生,并导入到数据库中,同时标记为僵尸网络流量.

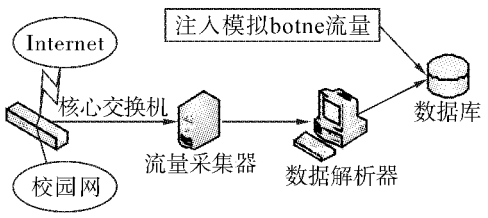


图 2　网络流量捕获结构

Fig. 2　Architecture of traffic capture

2.1.2　流的形成

主机间的一次 TCP 通信会形成一个 TCP 流,因此首先将捕获的原始 TCP 数据包转化为 TCP 流. 具有相同五元组(本地 IP 地址、本地端口号、远程 IP 地址、远程端口号、协议)的数据包集合构成一个流,同时满足前后相邻的 2 个数据包的时间间隔小于阈值 T_{in}(实验中 $T_{\text{in}}=60$ s). 其中本地 IP 地址和本地端口号是被测网络内主机的 IP 地址和端口号,远程 IP 地址和远程端口号是被测网络外的主机的 IP 地址和端口号.

一个 TCP 流通常以 3 次握手(SYN,SYN-ACK,ACK)开始,4 次握手(FIN,ACK,FIN,ACK)或者 RST 包结束. 因此首先按照五元组对数据包分组,组中按照时间排序,每组从第 1 个数据包开始扫描,找出 TCP 3 次握手包作为 TCP 流的开始,TCP 4 次握手包或者 RST 包作为 TCP 流的结束,从开始到结束的所有数据包组成一个 TCP 流. 记录每个 TCP 流,包括如下属性:本地 IP 地址、本地端口号、远程 IP 地址、远程端口号、流的开始时间、流的结束时间、流中数据包个数和流的字节数.

2.1.3　特征提取及预处理

根据第 1 节所述,以主机对为分析对象,提取主机对通信的流的特征,每个主机对有 6 个特征,这样,即可用下维向量表示一个主机对.

为了提高僵尸网络的检测率,在训练神经网络分类器前,需要对特征值进行预处理. 2.1.2 中提取出的主机对的各个特征值的取值范围差别很大,如果直接用提取出的特征向量作为神经网络输入,则训练的效果会受取值较大的特征值所影响,因此要对特征值归一化处理. 归一化操作将所有特征值映射到相同的范围内,能有效提高神经网络分类器的分类性能. 本文采用 z-score 方法进行规范化:

$$x' = \frac{x - \bar{x}}{\sigma_x}. \tag{1}$$

式中:x 是某一特征的值,$\bar{x}$ 和 σ_x 是该特征的平均值和标准差.

图 3 为特征值预处理的结果,前 2 列是主机对的 IP 地址,即内部主机和远程主机的 IP 地址,第 3 ~8 列为主机对的特征值,最后一列是类标签,0 表示非僵尸主机对,1 表示僵尸主机对. 从图 3 可以看出,预处理后所有特征值的取值都在同一个范围内.

localip	remoteip	flowCount	avgTimeInterval	avgTimeIntervalChange	avgDuration	avgPacketCount	avgFlowSize	isBot
211....	125.39...	16.00000	3153.552093	5428.852213	0.230163	8.000000	1528.875000	0
211....	116.57...	71.00000	21.589634	13.965107	1.981030	6.070423	1052.563380	0
211....	60.210...	35.00000	111.133755	113.041365	5.023963	6.371429	2520.114286	0
211....	121.19...	16.00000	240.387045	514.940462	0.237216	4.562500	1498.250000	0
211....	211.60...	11.00000	2193.627089	2437.764575	99.852817	83.454545	36743.27...	0
211....	124.89...	18.00000	2262.286818	2510.772981	1.156846	7.333333	3973.611111	0
211....	222.20...	141.00000	271.910141	521.469861	10.174067	12.581560	5353.361702	0
211....	119.42...	21.00000	1339.695635	2619.180881	0.235639	5.809524	1689.095238	0
211....	123.12...	49.00000	922.871677	1453.988139	0.011868	8.000000	849.612245	0
211....	61.55....	4549.0...	4.312675	7.509971	3.793326	6.244010	623.618158	0

(a) 预处理前

localip	remoteip	flowCount	avgTimeInterval	avgTimeIntervalChange	avgDuration	avgPacketCount	avgFlowSize	isBot
211....	125.39...	-0.024...	3.1293585539...	2.955594713409736	-0.00043...	-0.57729374...	-0.52704...	0
211....	116.57...	-0.003...	-0.673782092...	-0.5881145953913604	-0.00035...	-0.57437910...	-0.54700...	0
211....	60.210...	-0.016...	-0.565029630...	-0.5228570328072238	-0.00022...	-0.47914599...	-0.48303...	0
211....	121.19...	-0.024...	-0.408078227...	-0.2601270830930179	-0.00043...	-0.63678840...	-0.52830...	0
211....	211.60...	-0.025...	1.9637280590...	0.9987431956422216	0.003611...	3.04223962...	1.021036...	0
211....	124.89...	-0.023...	2.0471011075...	1.0465417018918	-0.00039...	-0.52211029...	-0.51949...	0
211....	222.20...	-0.0231...	-0.369999043...	-0.2558522946112179	0.000352...	-0.30409880...	-0.35884...	0
211....	119.42...	-0.022...	0.9368048210...	1.2484559267329	-0.00043...	-0.58517708...	-0.51991...	0
211....	123.12...	-0.011...	0.4206582841...	0.3545866130002936	-0.00044...	-0.57729374...	-0.56561...	0
211....	61.55....	1.8661...	-0.694741751...	-0.5923411225002294	-0.00027...	-0.56719475...	-0.56675...	0

(b) 预处理后

图 3　数据预处理结果

Fig. 3　The results of data preprocessing

2.2　神经网络分类器结构的设计

首先设计神经网络层次结构,主要是隐层的层

数,然后设计各层神经元个数,包括输入层、隐层、输出层的神经元个数,最后是神经网络的激活函数.

本文的神经网络分类器采用单隐层结构,因为如果隐层较多,神经网络的训练会比较慢,尤其数据量比较大时,单隐层有较快的训练速度,且不会对网络精度影响很大.考虑到网络流量一般较大,本文的神经网络模型采用单隐层.

本文的单隐层 BP 神经网络模型结构如图 4 所示.对应主机对的特征向量是六维的,因此输入层设置了六个神经元,分别是:1)TCP 流个数 flowCount;2)前后 2 个 TCP 流的时间间隔的平均值 avgInterval;3)时间间隔变化的平均值 avgItvChange;4)TCP 流的字节数平均值 avgByte;5)TCP 流的数据包个数平均值 avgPktCount;6)TCP 流的持续时间平均值 avgDuration.

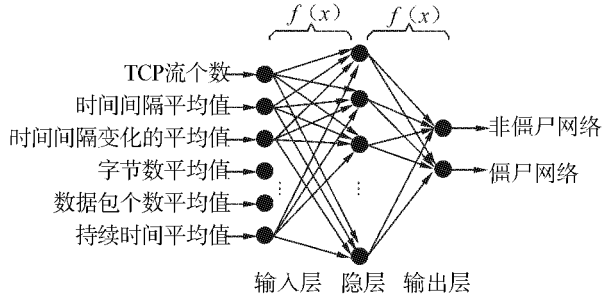


图 4 单隐层 BP 神经网络模型结构

Fig. 4 Model structure of BP neural network with a hidden layer

隐层神经元个数根据式(2)可取 3 ~ 12 的值,实验中对不同隐层神经元个数进行测试,选取最优值.式(2)为文献[12]指出的最佳隐层神经元个数,其中,k 为隐层神经元个数,m 为输入层神经元个数,n 为输出层神经元个数,α 为 1 ~ 10 的常数.

$$k = \sqrt{m+n} + \alpha. \tag{2}$$

输出层的神经元个数设置为 2.第 1 个输出神经元用于表示非僵尸主机对,第 2 个输出神经元用于表示僵尸主机对.如果是僵尸网络主机对,则第 1 个神经元的输出为 0,第 2 个神经元输出为 1;如果不是僵尸网络主机对,则第 1 个神经元输出为 1,第 2 个神经元输出为 0.实际训练过程中,输出神经元的取值是[0,1].训练时,若第 1 个神经元的值大于第 2 个,则将第 1 个神经元取值置为 1,将第 2 个神经元取值置为 0,并标记为非僵尸网络主机对.若第 1 个神经元的值小于第 2 个,则将第 1 个神经元取值置为 0,将第 2 个神经元取值置为 1,并标记为僵尸网络主机对.

BP 神经网络的激活函数采用 S 型函数,如式(3):

$$f(x) = 1/(1 + e^{x}). \tag{3}$$

2.3 神经网络分类器的训练流程

神经网络分类器设计完成后,即可利用训练数据对神经网络进行训练.BP 神经网络的训练过程是不断迭代的过程,其训练算法流程图如图 5 所示.

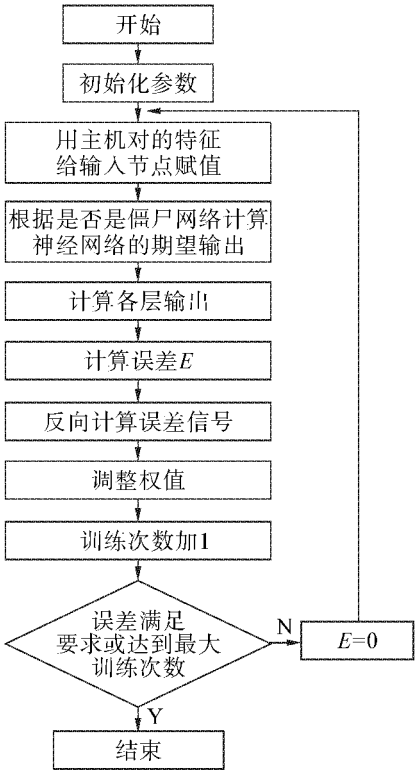


图 5 BP 神经网络分类器训练算法流程

Fig. 5 Flow chart of BP neural network training algorithm

具体步骤如下:

1)初始化神经网络所需的各个参数.参数包括输入层到隐层的权值和偏置、隐层到输出层的权值和偏置、权值和偏置的初始值是[-0.05,0.05]的随机值.另外,还需要设置神经网络的期望误差和最大训练次数.本文设置的期望误差是 0.001,最大循环次数是 1 000,即在 1 000 次循环内,误差小于等于 0.001,则 BP 神经网络分类器训练结束,否则循环 1 000 次后结束.

2)从训练样本中选 1 组主机对,给神经网络的输入节点赋值为主机对的 6 个特征.

3)计算神经网络的期望输出,如果是僵尸网络主机对,则输出层的第 1 个节点赋值为 0,第 2 个节点赋值为 1,如果不是僵尸网络主机对,则输出层的第 1 个节点赋值为 1,第 2 个节点赋值为 0.

4)计算出输入层到隐层再到输出层各节点的输出.

5)计算误差 E.

6)反向计算误差信号.

7)调整神经网络的权值和偏置.

8)重复 2) ~7),直到达到期望误差或者达到最大训练次数为止.

训练结束后,BP 神经网络分类器已形成,即可用训练好的分类器检测僵尸网络.

3　实验及结果分析

本文实验数据采集了天津某高校校园网内的网络流量,通过核心交换机端口镜像捕获网络流量,选择其中一个端口,被测子网内主机约 200 台,白天的网络流量为 150 ~200 MB/s. 实验中采用了 2011 年 4 月 6 日的网络流量,共 24 h. 真实网络环境中的流量作为背景流量,将其解析后导入数据库中,并注入模拟的僵尸网络流量. 随机选取背景流量中若干台主机,替换僵尸主机的 IP 地址为这些主机的 IP 地址,使这些主机既有正常流量也有僵尸网络流量,更符合真实场景. 本文模拟的僵尸网络流量特征符合现有大部分僵尸网络流量特征,因而本文研究这类僵尸网络,其他类型的僵尸网络不在本文研究范围内. 实验用 C#编写程序,用 SQL Server 2000 数据库存储解析后的网络流量数据.

为了确定最佳隐层神经元个数,做了如下实验. 分别将隐层神经元个数设置为 3 ~12 的不同值,观察训练误差和训练次数的变化情况. 实验中期望误差为 0.001,隐层神经元个数取不同值时神经网络训练误差都相同,因此只观察训练次数的变化. 图 6 是不同隐层神经元个数的训练循环次数,从图中可以看出训练循环次数随着隐层神经元个数增加而增大. 在误差相同情况下,训练次数越少越好,因此隐层神经元个数为 3 时效果最佳,后续的实验中都将隐层神经元个数设为 3.

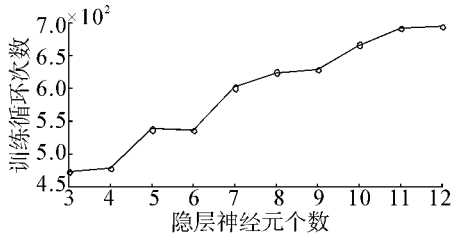


图 6　不同隐层神经元个数的训练循环次数

Fig. 6　The training times for different number of hidden layer neuron

图 7 是期望误差取不同值时的训练循环次数,实验数据取 3 次实验的平均值. 从图中可以看出,期望误差越大,训练循环次数越小. 因此期望误差大只需较少次数的循环误差即可达到期望值,可提高神经网络训练效率. 但是期望误差大,神经网络分类器的精度就下降,因此要取一个合理的值,使神经网络分类器的精度和训练次数达到平衡,从图 7 可以看出,期望误差为 0.002 时效果较好.

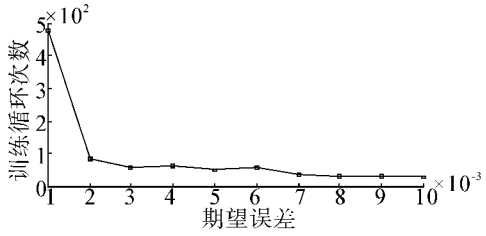


图 7　期望误差不同时的训练循环次数

Fig. 7　The training times for different expected error

为了验证神经网络分类器检测僵尸网络的性能,本文用测试数据对神经网络分类器进行测试,测试数据包含类标签,即是否是僵尸网络主机对. 本文采用检测率(detection rate,DR)和误报率(false positive rate,FPR)2 个指标对神经网络分类器的性能进行评估. 检测率为僵尸主机被正确检测出来的概率:

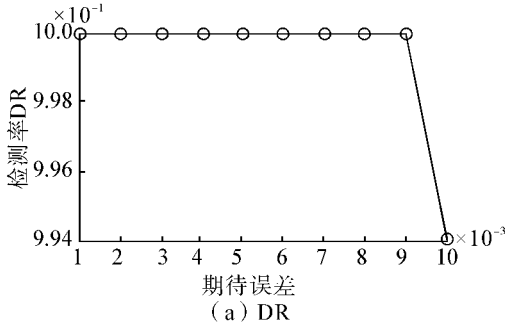
$$DR = \frac{TP}{TP + FN}. \tag{4}$$

式中:TP 表示正确检测出的僵尸主机个数,FN 表示未被检测出的僵尸主机个数. 误报率为合法主机被错误地检测为僵尸主机的概率,如式(5)

$$FPR = \frac{FP}{FP + TN}. \tag{5}$$

式中:FP 为被错误地检测为僵尸主机的合法主机个数,TN 为正确检测出的合法主机个数.

验证时首先以测试数据作为神经网络的输入,判断神经网络的检测结果和真实结果是否相同,然后分别统计 TP、FN、FP 和 TN 的值,最后根据式(4)和(5)计算检测率和误报率. 图 8 为期望误差不同时算法的检测率和误报率. 从图 8(a)可以看出,期望误差为 0.001 到 0.009 时,检测率都为 100%,当期望误差为 0.01 时,检测率下降到 99.4%.



(a) DR

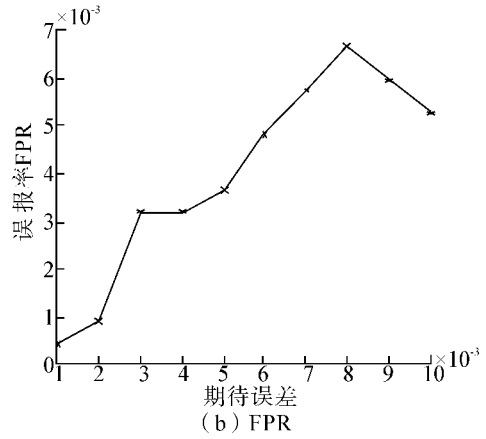


图8 期望误差不同时的检测率和误报率

Fig. 8 DR and FPR for different expected errors

从图8(b)可以看出,随着期望误差的增大,误报率呈上升趋势,期望误差为0.01时误报率稍微降低但检测率也降低.

综上所述,期望误差为0.001时检测率和误报率都达到最佳水平.从实验中也可以看出本文算法的检测率达到99%,误报率在1%以下,具有较好的检测性能.

4 结束语

本文提出了一种基于BP神经网络的僵尸网络检测方法.该方法以主机对作为分析对象,提取主机对之间通信的流量特征,然后将预处理后的特征作为神经网络的输入,训练BP神经网络分类器,用训练好的分类器进行僵尸网络检测.通过实验确定了最佳的BP神经网络隐层神经元个数和期望误差,实验表明,该方法有较高的检测率和较低的误报率,具有良好的性能.

参考文献:

[1]金鑫,李润恒,甘亮,等. 基于通信特征曲线动态时间弯曲距离的IRC僵尸网络同源判别方法[J]. 计算机研究与发展, 2012, 49(3): 481-490.
JIN Xin, LI Runheng, GAN Liang, et al. IRC botnets' homology identifying method based on dynamic time warping distance of communication feature curves[J]. Journal of Computer Research and Development, 2012, 49(3): 481-490.

[2]江健, 诸葛建伟, 段海新,等. 僵尸网络机理与防御技术[J]. 软件学报, 2012, 23(1): 82-96.
JIANG Jian, ZHUGE Jianwei, DUAN Haixin, et al. Research on botnet mechanisms and defenses[J]. Journal of Software, 2012, 23(1): 82-96..

[3] GOEBEL J, HOLZ T. Rishi: identify bot contaminated hosts by irc nickname evaluation[C]//Proceedings of USENIX First Workshop on Hot Topics in Understanding Botnets, Cambridge, USA, 2007: 1-12.

[4] FRANCOIS J, WANG S, STATE R, et al. BotTrack: tracking botnets using NetFlow and PageRank[M]//Lecture Notes in Computer Science. Valencia, Spain, 2011: 1-14.

[5] NAGARAJA S, MITTAL P, HONG C, et al. BotGrep: finding P2P bots with structured graph analysis[C]//Proceedings of the 19th USENIX Conference on Security. Washington, DC, USA, 2010: 1-16.

[6] GU G, PERDISCI R, ZHANG J, et al. BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection[C]//Proceedings of the 17th Conference on Security Symposium. San Jose, USA, 2008: 139-154.

[7] GU G, PORRAS P, YEGNESWARAN V, et al. BotHunter: detecting malware infection through IDSdriven dialog correlation[C]//Proceedings of the 16th USENIX Security Symposium. Boston, USA, 2007: 167-182.

[8] PRASAD K, REDDY A, KARTHIK M. Flooding attacks to internet threat monitors (ITM): modeling and counter measures using botnet and honeypots[J]. International Journal of Computer Science and Information Technology, 2011, 3(6): 159-172.

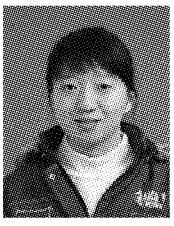
[9] ZHANG J, PERDISCI R, LEE W, et al. Detecting stealthy P2P Botnets using statistical traffic fingerprints[C]//Proceedings of IEEE/IFIP 41st International Conference on Dependable Systems and Networks. Hong Kong, China, 2011: 121-132.

[10]方滨兴,崔翔,王威. 僵尸网络综述[J]. 计算机研究与发展, 2011, 48(8): 1315-1331.
FANG binxing, CUI Xiang, WANG Wei. Survey of botnets[J]. Journal of Computer Research and Development, 2011, 48(8): 1315-1331.

[11] WANG P, WU L, ASLAM B, et al. A systematic study on peer-to-peer botnets[C]//Proceedings on Computer Communications and Networks. San Francisco, USA, 2009: 1-8.

[12]飞思科技产品研发中心. 神经网络理论与MATLAB7实现[M]. 北京: 电子工业出版社, 2005: 1-108.

作者简介:



蒋鸿玲,女,1986年生,博士研究生,主要研究方向为网络安全与云计算等,发表学术论文7篇.

邵秀丽,女,1963年生,教授,博士生导师,主要研究方向为云计算与软件工程等,发表学术论文80余篇.