SDN 环境下基于 BP 神经网络的 DDoS 攻击检测方法*

王晓瑞[1], 庄 雷[1†], 胡 颖[1], 王国卿[1], 马 丁[1,2], 景晨凯[1]
(1. 郑州大学 信息工程学院, 郑州 450001; 2. 河南工业大学 信息科学与工程学院, 郑州 450001)

摘 要: 软件定义网络是一种全新的网络架构,集中控制是其主要优势,但若受到 DDoS 攻击则会造成信息不可达,也容易造成单点失效。为了有效地识别 DDoS 攻击,提出了一种 SDN 环境下基于 BP 神经网络的 DDoS 攻击检测方法。该方法获取 OpenFlow 交换机的流表项,分析 SDN 环境下 DDoS 攻击特性,提取出与攻击相关的流表匹配成功率、流表项速率等六个重要特征;通过分析六个相关特征值的变化,采用 BP 神经网络算法对训练样本进行分类,实现对 DDoS 攻击的检测。实验结果表明,该方法在有效提高识别率的同时,降低了检测时间。通过在软件定义网络环境中的部署,验证了该方法的有效性。

关键词: 软件定义网络; 分布式拒绝服务攻击; 反向传播神经网络; 特征值; 攻击检测
中图分类号: TP393.08 **文献标志码**: A **文章编号**: 1001-3695(2018)03-0911-05
doi:10.3969/j.issn.1001-3695.2018.03.057

DDoS attack detection based on BPNN in software defined networks

Wang Xiaorui[1], Zhuang Lei[1†], Hu Ying[1], Wang Guoqing[1], Ma Ding[1,2], Jing Chenkai[1]
(1. *School of Information Engineering, Zhengzhou University, Zhengzhou* 450001, *China*; 2. *College of Information Science & Engineering, Henan University of Technology, Zhengzhou* 450001, *China*)

Abstract: Software definition network is a new network architecture that achieves a centralized network control. Although centralized control is the main advantage of SDN, but if subject to DDoS attacks, the information will be not reachable, it also likely to cause a single point of failure. In order to mitigate this threat, this paper proposed a DDoS attack detection method based on SDN centralized control. This algorithm obtained the flow table items of OpenFlow switch, analyzed the characteristics of DDoS attacks in SDN environment, and extracted six characteristics related to attacks. By analyzing the changes of the six eigenvalues, it used BP neural network algorithm to classify the training samples to achieve the DDoS attack detection. The experimental results show that the method can improve the recognition rate and reduce the detection time. The effectiveness of the method is verified by the deployment in a software-defined network environment.

Key words: software definition network; DDoS attack; BP neural network; eigenvalues; attack detection

0 引言

SDN(software define network)是一种新型的网络体系架构,实现了网络控制平面和数据平面的分离。与传统的网络体系架构相比,SDN 在可编程性、硬件通用性和管理控制方式等方面具有明显的优势。其控制平面主要由控制器组成,控制器负责连接底层交换设备与上层应用;数据平面由交换机实现,主要负责数据的高速转发[1]。随着 SDN 的广泛应用,SDN 的安全问题引起了广泛的关注。在 SDN 体系结构中,因为控制平面和数据平面相解耦,一旦交换机和控制器之间的连接失败,整个网络将失去控制,因此控制器的安全是整个 SDN 网络安全保障的关键之一。控制器安全的主要威胁之一是分布式拒绝服务(distributed denial of service,DDoS)攻击。DDoS 攻击是指攻击者通过傀儡主机,消耗攻击目标的计算资源,防止目标主机为合法用户提供服务。在 DDoS 攻击中,攻击者首先侵入 SDN 中的部分主机,通过向网络中输入大量伪造的无效网络流量,最终导致控制器资源耗尽,并造成合法的数据包无法完成转发。为防范攻击者对交换机的非授权访问,避免控制器被攻击者非法控制,如何快速准确地检测 DDoS 攻击是 SDN 安全领域研究的热点。

DDoS 攻击是当今互联网最重要的威胁之一。目前针对传统网络架构,研究者提出了大量的 DDoS 攻击检测方法,如文献[2]从数据包级和会话流级进行分析,采用支持向量机(SVM)分类器建立 DDoS 攻击检测模型,提出一种基于 PCC(packet and conversation considering with context)时间序列的检测算法,全面地描述 DDoS 攻击过程。文献[3]将流量和 IP 熵特性结合起来,与单一通过流量或 IP 熵检测攻击相比,更准确地检测出了 DDoS 攻击。文献[4]分析 TCP/UDP/ICMP 的不同,基于每个协议特定的特征通过训练 ANN 算法去检测 DDoS 攻击。但是 SDN 是一种新型的网络,工作原理和传统网络不同,随着 SDN 网络领域的广泛应用,研究人员对 SDN 的网络安

收稿日期: 2016-12-02; **修回日期**: 2017-02-16 **基金项目**: 国家“973”计划资助项目(2012CB315901);国家自然科学基金资助项目(61379079);河南省科技攻关项目(122102210042)

作者简介:王晓瑞(1990-),女,河南安阳人,硕士,主要研究方向为软件定义网络;庄雷(1963-),女(通信作者),山东日照人,教授,博士,主要研究方向为网络虚拟化、下一代互联网(ielzhuang@zzu.edu.cn);胡颖(1982-),女,河南商丘人,讲师,博士,主要研究方向为网络功能虚拟化、下一代互联网;王国卿(1989-),男,山东临沂人,博士研究生;马丁(1978-),男,河南夏邑人,讲师,博士,主要研究方向为网络功能虚拟化、互联网体系结构;景晨凯(1991-),男,河南登封人,硕士,主要研究方向为深度学习.

全也进行了研究。文献[5]提取流表的关键特征,采用 KNN 算法检测 DDoS 攻击,与 OpenFlow 技术结合在一起,通过分析流表中的五个特征值有效地检测 DDoS 攻击,但特征数据选取时忽略了流表项速率的重要性,攻击发生时,流表项请求数目也会在固定时间内增加。文献[6]提出了基于合法的源和目标 IP 地址数据库的 DDoS 攻击检测机制,利用 DDoS 攻击发生时源和目的 IP 地址异常的特性有效地检测出 DDoS 攻击,但 OpenFlow 技术主要是基于流进行数据转发,可以通过获取全网信息简单有效地进行攻击检测和缓解。文献[7]采用支持向量机对训练样本进行分类,实现 DDoS 攻击的检测。将该 DDoS 攻击检测方法进行原型系统实现并集成到 SDN 环境中,验证了该方法的正确性和有效性,但在选取流量特征时忽略了端口增速的重要性。DDoS 攻击时会随机生成端口号,端口号会剧增。基于以上分析,本文提出了一种 SDN 环境下基于 BP 神经网络的 DDoS 攻击检测方法。该方法通过分析 DDoS 攻击特点和 OpenFlow 技术,提出了更全面的六个流表特征值如流表项匹配成功率等,并利用 BPNN 分类算法检测攻击。在攻击检测时设置优先级,根据攻击检测优先级检测攻击,对检测结果为攻击的进行攻击处理。BPNN(back-propagation neural network,反向传播神经网络)非线性分类能力较强,能很好地处理线性不可分的问题。文献[4]采用 BPNN 算法基于特定的特征去实时检测传统网络中已知和未知的 DDoS 攻击,并且检测率得到有效的提高。

1　SDN 环境下的 DDoS 攻击

SDN 网络将控制平面和数据平面分开,当数据平面收到一个新的网络数据包时,必须从控制平面获取流规则,并按照流规则处理数据包。这种被动控制模式使网络运营商能够有效地控制网络。然而,它也带来了一些安全性的问题。在匹配时一旦和流表项匹配不成功,则会发送请求至控制器。从图 1 所示的流程图可以看出,不存在安全机制来检查流和在数据平面所有不匹配数据包的合法性,都将被发送到控制器。

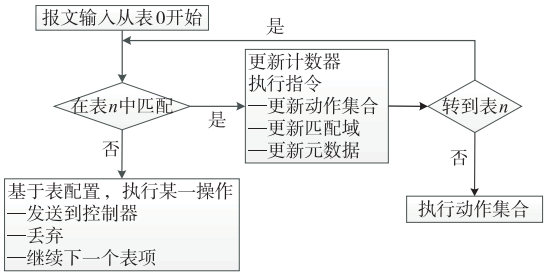

图1　匹配流程

SDN 有两个潜在的安全威胁[8]:

a)数据平面。攻击者主机产生的虚假通信请求将产生许多无用的流条目,占用很多流表空间。因为流表空间是有限的,当流表溢出时合法的请求将被拒绝。

b)控制平面。源 IP 地址欺骗是 DDoS 攻击广泛采用的技术。当在数据平面受到 DDoS 攻击时,会出现不匹配的流的请求到控制器。控制器的计算和存储资源将由这些假请求占用。

2　基于 BPNN 的 DDoS 攻击检测方法

BPNN 攻击检测方法包括五个模块,分别为流表收集模块、特征提取模块、数据训练模块、攻击检测模块、攻击处理模块,如图 2 所示。流表收集模块定期向 OpenFlow 交换机发送流表请求,交换机回复的流表信息通过加密信道传给流表收集模块。特征提取模块负责从流表收集模块收集的流表中提取出与 DDoS 攻击相关的特征值。数据训练模块使用 BPNN 算法将收集的特征值信息进行训练。攻击检测模块通过对比网络数据包和训练结果后判断,当前网络是否受到攻击。若检测到攻击则由攻击处理模块通知控制器,对攻击进行处理。

图2　攻击检测流程

2.1　流表收集

收集流表信息主要通过 OpenFlow 协议来实现,如图 3 所示。交换机回复控制器定期发送的 ofp_flow_stats_request 报文,获得流表的时间间隔应适中,设置流表周期获取时间与 OVS 控制器设置的近期未命中流删除时间保持一致。

```
root@jck-Lenovo:~# ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=30.555s, table=0, n_packets=1, n_bytes=42, idle_timeout=60
, idle_age=30, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=fe:2e:60:2d:3
9:12,dl_dst=ba:b9:46:bd:71:ef,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions
=output:2
 cookie=0x0, duration=35.559s, table=0, n_packets=1, n_bytes=42, idle_timeout=60
, idle_age=35, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=ba:b9:46:bd:7
1:ef,dl_dst=fe:2e:60:2d:39:12,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions
=output:1
```

图3　流表收集

执行 sudo ovs-ofct1 dump-flows s1 > a. txt,然后执行 cat a. txt。读取重定向文件,进行流表收集。

2.2　流特征提取

DDoS 攻击者的攻击方式虽然是多样的,但是大多数攻击流量具有一定的规律。因此可以通过流表项信息分析单位时间内网络流量分布特性的变化,从而检测攻击。根据 OpenFlow 协议,交换机的流表是数据包的转发依据,每个流表由多个流表项组成。流表项是数据转发的规则,结构如图 4 所示。

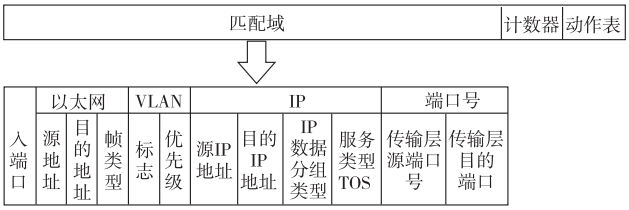

图4　流表项结构

流表项中的计数器用于统计数据流的基本信息,如匹配该流表项的数据包数、查找次数、收发分组数、生存时间等[1]。

DDoS 攻击时大量的傀儡机被控制同时发起攻击,攻击者会随机伪造攻击数据包源 IP 地址,因而源 IP 地址更加分散,且数量巨大;网络数据包集中流入受害机,向提供服务的某些端口发起攻击,所以网络数据包的目的 IP 地址、目的端口地址会相当集中。正常状态下的数据具有较高的相似性,而在 DDoS 攻击过程中,流量特征突变,大多数攻击流量具有一定的规律,因此,通过获取流表项信息可以分析单位时间内网络流量分布特性的变化,从而检测攻击流。基于以上分析,提取流表项相关信息并转换为有关 DDoS 攻击的一维特征信息,包括以下六项:

a)流包数均值(average number of packets in per flow,APF)。

正常状态和受到攻击时的流包数是不同的。在攻击中通常是连续地、随机地生成假 IP,所以流的生成速度加快,且每个流减小包量。例如,一般每个流有 1 ~3 个数据包。

$$APF = \left(\sum_{j=1}^{FlowNum} PacketsNum_j\right)/FlowNum \tag{1}$$

其中:$PacketsNum_j$ 是在一定时间间隔内 j 流中数据包的数目;FlowNum 是该时间间隔内所有数据包的数目。

b)对流比(percentage of pair-flow,PPF)。

正常状态下的流是为了获取或者提供服务,所以具有交互性。在攻击状态下因为是伪造的源 IP 地址,所以无法提供正常的服务。

$$PPF = \frac{2 \times Pair - FlowsNum}{FlowsNum} \tag{2}$$

其中:Pair 是交互流的对数;FlowsNum 是流的总数。

c)端口增速(port generating speed,PGS)。

网络处在正常状态下端口增速相对稳定。DDoS 攻击时会随机生成端口号,所以在攻击发生时端口的增速会明显增大。

$$PGS = \frac{PortNum}{interval} \tag{3}$$

其中:interval 为采样周期。

d)源 IP 增速(source IP growing speed,SIS)。

$$SIS = \frac{sIPNum}{interval} \tag{4}$$

DDoS 的主要攻击特点就是源 IP 欺骗,伪造源 IP 地址发送大量的数据包。这一特性使得发生攻击时源 IP 地址的增速会在固定时间显著增加,所以可以将源 IP 增速作为攻击特征的属性之一。

e)流表项速率(rate of flow entries,rFE)。

$$rFE = \frac{FlowsNum}{interval} \tag{5}$$

攻击发生时,将增加网络中针对特定主机的请求,导致有关该主机的流表项请求数目在固定时间内也有所增加,因此,可以通过流表项速率表征攻击属性。

f)流表匹配成功率(match/lookup ratio,Mlr)。

$$Mlr = \frac{Match}{lookup} \tag{6}$$

每当数据包到达 OpenFlow 交换机时,OpenFlow 交换机都将会执行查找和匹配操作,它与新流的数量相关。当流表溢出并且大多数流是新流时,匹配成功率将急剧减少。

2.3　分类算法

由于正常状态下和受到攻击时的流量特征是不同的,所以可以把攻击检测问题看成分类问题,通过特征值来判断网络状态是正常或是异常,从而将正常状态和攻击状态分开。攻击检测的基本处理流程为:选择合适的网络流特征即流包数均值、对流比、端口增速、源 IP 增速、流表项速率、流表匹配成功率,组成六元组样本特征序列,样本序列有正常和异常两种状态。选择机器学习算法中的 BP 神经网络算法构建检测模型,使用模型对未标记的特征样本序列进行分类。使用的 BP 神经网络结构如图 5 所示。

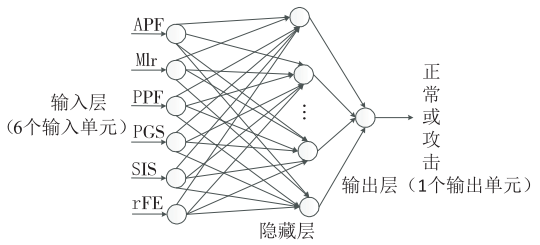

图 5　BPNN 网络结构图

BP 神经网络具有完备的理论体系和学习机制。其算法的主要思想是通过周而复始的正向传播与反向调节,不断修正神经元间的权值,当误差满足精度要求时,停止学习。反向传播算法流程如图 6 所示,步骤如下:

a)初始化网络权值。

b)重复以下步骤直至收敛(对各样本依次计算)。

(a)从前向后各层计算各单元 O_j,使误差沿网络反向传播。

$$net_j = \sum_i w_{ij} O_i \tag{7}$$

$$O_j = 1/(1 - e^{-net_j}) \tag{8}$$

(b)对输出层计算 δ_j,计算出网络的每个输出单元的误差项。

$$\delta_j = (y - O_j) O_j (1 - O_j) \tag{9}$$

(c)从后向前计算各隐层 δ_j,计算出网络的每个隐藏单元的误差项。

$$\delta_j = O_j (1 - O_j) \sum \omega_{jk} \delta_k \tag{10}$$

(d)通过计算并保存各权值修正量。

$$\Delta\omega_{ij}(t) = \alpha \Delta\omega_{ij}(t-1) + \eta \delta_j O_j \tag{11}$$

(e)由步骤(d)得到的权值修正量修正权值,从而逐步逼近目标输出。

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \Delta\omega_{ij}(t) \tag{12}$$

其中:O_j = 单元 j 计算出的输出,net_j = 单元 j 的输入加权和;δ_j 表示与单元 j 相关联的误差项;$\alpha\Delta\omega_{ij}(t-1)$ 为冲量函数。其中 α 是一个称为冲量的常数,作用是增加冲量,使一次迭代到下一次迭代时以同样的方向滚动,它也具有在梯度不变的区域逐渐增大搜索步长的效果,从而可以加快收敛;$\Delta\omega_{ij}(t)$ 是算法主循环中的第 t 次迭代进行的权值更新。

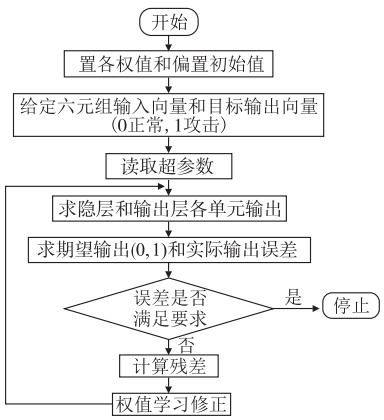

图6　BPNN算法流程

2.4　攻击检测

设定 threshold(APF,PPF,PGS,SIS,rFE,Mlr)为触发策略,分别用 a_1、a_2、a_3、a_4、a_5、a_6 表示;(y_1、y_2、y_3、y_4、y_5、y_6)表示测试集。设置初值 alarm = 0。当任意一个特征值超过阈值时认为是攻击的可疑点,即 $y_i > a_i$,并设置 alarm = 1,激活 BPNN 攻击检测方法。这样在保证不减少检测率的情况下也减轻了全部检测带来的负载问题。其中阈值选取的依据是异常数据的特征值。对每个特征值进行统计,并对其进行归一化处理,统计其区间。从区间内选取 PR 曲线上最接近对角线的点作为阈值,该点准确率与召回率约为 1:1。

设攻击检测优先级 $P = i$,超过阈值的特征值的总数为 count,初始值为 0;若 $y_i > a_i$,则 count + 1。其中 count,i 的取值

如表 1 所示。

表 1　攻击检测优先级表

count	1	2	3	4	5	6
i	6	5	4	3	2	1

2.5　攻击处理

若攻击检测结果是攻击,则将 alarm 设置为 2,向控制器发出攻击警告。控制器接收预警,下发指令至防火墙,更改防火墙配置,减少攻击流量,同时下发流表更改指令至 OpenFlow 交换机,更改流表匹配项设置,并通过 meter 表控制流表速率,丢弃攻击数据包。

一个新流到达时,OpenFlow 交换机将给控制器发送一个 PACKET_IN 事件。PACKET_IN 事件的数量与新流的数量具有正相关性。PACKET_IN 事件越多,控制器的负载越重。所以在此将 PACKET_IN 事件作为控制器处理攻击时优先级设定的考虑因素。PACKET_IN 事件越多,则优先级越高,控制器优先处理优先级高的攻击。

3　实验及分析

3.1　实现环境

通过部署软件定义网络环境来验证本文 DDoS 攻击检测方法的有效性。在 Ubuntu 环境下部署三台 OpenFlow 交换机。其中内核级虚拟化(kernel-based virtual machine,KVM)作为终端主机。实验采用 ODL(Open Daylight)控制器和 OVS(open vSwitch)交换机。网络拓扑图如图 7 所示。

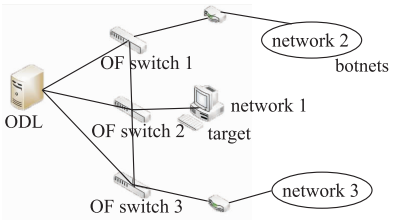

图7　网络拓扑图

图 7 中,network1 对应于受害者网络,它是由 ODL 控制器控制的三个 OpenFlow 的交换机;network2 是产生的 DDoS 洪水流量的僵尸网络。反向散射通信和 IP 欺骗流向放置在 network2 和 network3 主机。训练样本生成阶段中,正常流量由 network1 中主机进行正常访问而产生,其中包括 TCP 流量、UDP 流量、ICMP 流量等;CAIDA DDoS 2007 数据集作为异常流量。网络流量攻击类型包括 TCP SYN flood、UDP flood、ICMP flood 等。通过代码 sudo ovs-ofctl dump-flows s1 查看流表,再对上述流量通过 OVS 交换机产生的流表进行收集,分为正常训练样本和 DDoS 攻击流量训练样本,并转换为攻击检测的六个特征值。

3.2　结果分析

算法部分的实验用 Python 实现。选用 sigmoid 函数作为激活函数。最初选用二次代价函数作为损失函数,在第四次选用交叉熵函数作为代价函数与二次代价函数再进行对比。

输出层和输入层的单元数是由问题本身决定的,在本文中输入单元数是六特征维数,输出单元是 0 正常状态和 1 攻击状态,但中间隐层的单元数如何决定则缺乏有效的方法。一般来说,问题越复杂,需要的隐层单元越多;或者说同样的问题,隐层单元越多越容易收敛,但是隐层单元数过多会增加使用时的计算量,而且会产生过拟合问题,使对未出现的样本的推广能力变差。当隐层数难以确定时,先选较多的隐层单元数,待学习完成后,再逐步删除一些隐层单元,使网络更为精简。网络的每一个输入节点对应样本的一个特征。

本文将 5 000 个样本集分成三个部分:3 000 个训练集、1 000个验证集、1 000 个测试集。使用识别率(recognition rate,RR)这个评价指标对实验结果进行评估:

$$RR = (TP + TN)/(TP + TN + FP + FN) \times 100\%$$

其中:*TP* 表示被正确标记的正常测试样本数;*FP* 表示被错误标记的正常测试样本数;*TN* 表示被正确标记的攻击测试样本数;*FN* 表示被错误标记的攻击测试样本数。

第一次实验。划分为三层,输入节点 6,隐藏层单元为 20,输出层为 2;偏置和权重使用均值为 0、标准差为 1 的高斯分布进行随机初始化。随机初始化给随机梯度下降算法一个起点,激活函数选择 sigmoid 函数。在每个迭代期,它首先随机地将训练数据打乱,然后将其分成多个适当大小的小批量数据。这是一个简单的从训练数据的随机采样方法。然后对于每一个 mini_batch 大小设为 20,本文应用一次梯度下降,学习速率设为 2.0,学习超过 20 epoches。经过训练的网络给出的识别率约为 95%(峰值时为 95.36%)。

第二次实验。将隐藏层单元设置为 50,其他保持不变,训练时间花费更长,它将结果提升至 96.09%。更多的节点拟合数据的能力也相应有所提高了,但是训练的时间也更长。在这种情况下,使用更多的隐藏神经元能得到更好的结果。

第三次实验。保持隐藏层数目为 50,选定学习速率为 0.001 时,前三个 eopch,识别率为 95.4%,识别率并没有提升;在 8 个 epoch 之后,识别率为 96.11%,有所提升,这表明增加学习速率可以提高识别率。此后逐渐增大学习速率,经过不断地尝试,最终设为 1.0。

第四次实验。选择交叉熵为代价函数,mini_batch 大小为 20,学习率为 0.5,隐藏层单元分别为 20 个和 50 个,识别率分别为 95.4% 和 96.23%,相比二次代价函数,略微有所提升。

第五次实验。使用 weight_decay = 2.0,mini_batch 大小为 20,学习率为 0.5,隐藏层单元分别为 20 个和 50 个,识别率分别为 96.32% 和 97.23%,学习率为 1.0,weight_decay = 3.0。识别率达到 97.34%。

第六次实验。使用两个隐藏层,第一个隐藏层 20 个单元,第二个隐藏层同样 20 个单元,学习率 1.0,,weight_decay = 3.0,识别率达到 98.94%;使用三个隐藏层,数目均为 20,其他保持不变,识别率 97.3%,没有发生明显改变,甚至出现了下降,这是由于梯度不稳定引起的,深层网络较难训练。

第七次实验。使用两个隐藏层,第一个隐藏层 20 个单元,第二个隐藏层 30 个单元,学习率 1.0,weight_decay = 3.0,识别率 99.47%。

所有实验结果如表 2 所示。其中:weight_decay 是权值衰减惩罚项使得权值收敛到较小的绝对值,而惩罚大的权值,因为大的权值会使得系统出现过拟合,降低其泛化性能。学习率为参数更新每一步的改变量,在每个 epoch 中,它通过随机地 shuffling 训练数据,然后把它分成合适大小的 mini_batches。对于每个 mini_batch,应用一次梯度下降算法,根据梯度下降算

法的一次迭代更新权值和偏置。七次实验识别率的变化如图 8 所示。

表 2　选择不同参数时 BPNN 对攻击检测的性能

参数	实验 BPNN						
	1	2	3	4	5	6	7
输入节点	6	6	6	6	6	6	6
隐藏层单元	20	50	50	50	50	20 + 20	20 + 30
输出层	2	2	2	2	2	2	2
学习速率	2.0	2.0	1	0.5	1.0	1.0	1.0
代价函数	二次代价函数	二次代价函数	二次代价函数	交叉熵函数	交叉熵函数	交叉熵函数	交叉熵函数
weight-decay	-	-	-	-	3.0	3.0	3.0
识别率/%	95.36	96.09	96.11	96.24	97.34	98.34	99.47

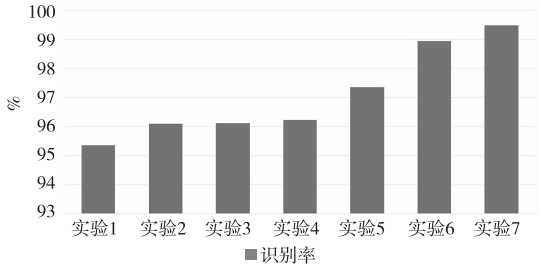

图8　实验对比图

最后在 SDN 中使用传统网络攻击检测的方法并用 SVM 算法进行检测,由文献[7]中各个核函数分类性能的对比可知性能最高的是线性核函数。所以在此核函数使用线性核函数,识别率为 94.6%、检测时间为 615 μs。接着再用本文中提取的六个特征进行攻击检测,经过调整后的 SVM 算法识别率为 98.7%、检测时间为 350 μs,如表 3 所示。

表 3　SDN 和传统网络中不同算法对 DDoS 攻击的检测结果

网络	算法	识别率/%	检测时间/μs
SDN	本文算法	99.47	225
SDN	调整后的 SVM 算法	98.7	350
传统网络	SVM 算法	94.6	615

实验表明 BPNN 算法使用两个隐藏层,隐藏单元分别为 20 和 30,学习率为 1.0,weight_decay 为 3.0 时识别率达到最优,并用最优结果的参数值去验证六个特征值的必要性,即将六个特征值选择一个特征值不参与 BPNN 算法的训练,将最后检测结果与六个特征值的检测结果进行对比,若识别率降低则说明该特征值有效,反之无效。经过对比实验,选取的六个特征值中没有冗余项。

4　结束语

本文提出了一种基于 SDN 架构的 DDoS 攻击检测方法,通过分析 DDoS 的攻击方式,充分利用控制器在 SDN 网络中集中控制的特点,实时获取 OpenFlow 交换机信息并提取出与 DDoS 紧密相关的六元组,结合 BPNN 算法处理流量的关键特征值从而检测 DDoS 攻击。该方法的优点在于较为全面地提取和分析 SDN 架构下流量的关键属性,并通过设置阈值减轻整个网络的负载。通过在软件定义网络环境中的部署,验证了该方法的有效性。

参考文献:

[1] 左青云, 陈鸣, 赵广松, 等. 基于 OpenFlow 的 SDN 技术研究[J]. 软件学报, 2013,24(5): 1078-1097.

[2] 郑亚, 陈兴蜀, 尹学渊. 基于 PCC 时间序列的 DDoS 检测算法[J]. 四川大学学报: 工程科学版, 2015(S2): 142-148.

[3] 杨君刚, 王新桐, 刘故膏. 基于流量和 IP 熵特性的 DDoS 攻击检测方法[J]. 计算机应用研究, 2016,33(4): 1145-1149.

[4] Saied A, Overill R E, Radzik T. Detection of known and unknown DDoS attacks using artificial neural networks[J]. Neurocomputing, 2016,172(C): 385-393.

[5] 肖甫, 马俊青, 黄洵松, 等. SDN 环境下基于 KNN 的 DDoS 攻击检测方法[J]. 南京邮电大学学报: 自然科学版, 2015,35(1): 84-88.

[6] Wang Xiulei, Chen Ming, Xing Changyou, *et al*. Defending DDoS attacks in software-defined networking based on legitimate source and destination IP address database[J]. IEICE Trans on Information & Systems, 2016, E99.D(4): 850-859.

[7] 李鹤飞, 黄新力, 郑正奇. 基于软件定义网络的 DDoS 攻击检测方法及其应用[J]. 计算机工程, 2016,42(2): 118-123.

[8] Wang Xiulei, Chen Ming, Wei Xianglin, *et al*. Defending DDoS attacks in software defined networking based on improved Shiryaev-Roberts detection algorithm[J]. Journal of High Speed Networks, 2015, 21(4): 285-298.

[9] Braga B R, Mota M E, Passito P A. Lightweight DDoS flooding attack detection using NOX/OpenFlow[C]//Proc of IEEE Conference on Local Computer Networks. Washington DC: IEEE Computer Society, 2010: 408-415.

[10] 王帅, 金华敏. 基于 SDN 的安全分析及关键技术研究[J]. 电信科学, 2014(S2):45-49.

[11] Scott-Hayward S, O'Callaghan G, Sezer S. SDN security: a survey[C]//Proc of IEEE SDN for Future Networks and Services. [S.l.]: IEEE Press, 2013: 1-7.

[12] Li Jin, Liu Yong, Gu Lin. DDoS attack detection based on neural network[C]//Proc of International Symposium on Aware Computing. 2010: 196-199.

[13] Mousavi S M, Sthilaire M. Early detection of DDoS attacks against SDN controllers[C]//Proc of International Conference on Computing, Networking and Communications. Washington DC: IEEE Computer Society, 2015: 77-81.

[14] 李鹤飞, 董晨, 郑晓航, 等. 基于软件定义网络的流量管理应用的研究和实现[J]. 计算机应用与软件, 2015,32(5): 17-19.

[15] 成亚男, 董晨, 褚灵伟, 等. 基于软件定义网络的防火墙系统设计与实现[J]. 计算机应用与软件, 2015,32(1): 286-288.

[16] Kokila R T, Thamarai Selvi S, Govindarajan K. DDoS detection and analysis in SDN-based environment using support vector machine classifier[C]//Proc of the 6th International Conference on Advanced Computing. 2014.

[17] 何亨, 黄伟, 李涛, 等. 基于 SDS 架构的多级 DDoS 防护机制[J]. 计算机工程与应用, 2016,52(1): 81-88.

[18] 张永铮, 肖军, 云晓春, 等. DDoS 攻击检测和控制方法[J]. 软件学报, 2012, 23(8): 2058-2072.

[19] Lim S, Yang S, Kim Y, *et al*. Controller scheduling for continued SDN operation under DDoS attacks[J]. Electronics Letters, 2015, 51(16): 1259-1261.

[20] 孙鹏, 刘秋妍. SDN 安全技术研究[J]. 中国电子科学研究院学报, 2015, 10(4): 416-420.

[21] 张朝昆, 崔勇, 唐罡罡, 等. 软件定义网络(SDN)研究进展[J]. 软件学报, 2015,26(1): 62-81.

[22] Feng Yifu, Guo Rui, Wang Dongqi, *et al*. Research on the active DDoS filtering algorithm based on IP flow[C]//Proc of International Conference on Natural Computation. 2009: 628-632.

[23] 左青云, 陈鸣, 王秀磊, 等. 一种基于 SDN 的在线流量异常检测方法[J]. 西安电子科技大学学报: 自然科学版, 2015,42(1): 155-160.

[24] Klaedtke F, Karame G O, Bifulco R, *et al*. Towards an access control scheme for accessing flows in SDN[C]//Proc of the 1st IEEE Conference on Network Softwarization. [S.l.]:IEEE Press, 2015.

[25] 黄韬, 刘江, 魏亮, 等. 软件定义网络核心原理与应用实践[M]. 北京: 人民邮电出版社, 2014.