

# [Archon-智能SDN网络监测与DDos防御系统] 设计文档

所在赛道与赛项: **B-EP1**

## 目录

1 目标问题与意义价值 .....	3
2 设计思路与方案 .....	4
2.1 设计思路 .....	5
2.2 技术路线 .....	5
2.3 主要设计方案 .....	5
2.3.1 SDN时延带宽监测设计方案 .....	6
2.3.2 SDN防火墙设计方案 .....	7
2.3.2 DDos入侵检测器设计方案 .....	7
2.3.3 管理系统设计方案 .....	8
3 方案实现 .....	9
3.1 SDN时延带宽监测实施方案 .....	9
3.1.1 OFPT_STATS_REQUEST&&REPLY消息 .....	10
3.1.2 EchoRequest&EchoReply消息 .....	11
3.1.3 回送消息处理与时延计算 .....	11
3.2 SDN防火墙实施方案 .....	11
3.2.1 Mininet工具 .....	12
3.2.2 Ryu控制器 .....	12
3.2.3 网络拓扑结构 .....	12
3.2.4 规则设计 .....	13
3.2.5 连接DDos入侵检测器 .....	15
3.3 DDos入侵检测器实施方案 .....	15
3.3.1 深度网络的构建及训练 .....	16
3.3.2 实时数据包的特征提取与分类 .....	31

3.4 管理系统实施方案 .....	33
4 运行结果/应用效果 .....	34
4.1 日志查看 .....	34
4.2 规则编辑 .....	35
4.3 系统测试 .....	36
5 创新与特色 .....	39
6 后续优化 .....	40

# 1 目标问题与意义价值

随着云计算越来越普及，人们更乐意在云平台部署自己的app应用，从而，云端资源成为了网络安全技术所要保护的重要对象。云防火墙、云网络的入侵检测系统是保护虚拟网络的重要防线。对于一个网络系统而言，分布式拒绝服务攻击(Distributed Denial of Service,DDoS)是一种最常见的威胁网络安全的攻击手段，其目的是让目标网络耗尽恶意流量。虽然很多统计方法已经被设计用于DDoS攻击检测，但设计一个计算开销低的实时检测器和在线时延带宽监测系统仍然是主要关注点之一，时延监测系统是判断系统是否异常的重要手段之一，现代网络防御系统所面临的攻击手段不再单一，所需要的防御技术也不再单一，入侵检测系统集成流量监测系统是非常有必要的。

链路时延检测和带宽占用检测是计算机网络领域一项经典的技术，带宽时延是判断一个网络是否正常最直观的指标，也是判断是否发生网络拥塞，链路崩溃最直观的指标，这也是DDos攻击最终的目的，所以，在防火墙和入侵检测系统发生作用的同时，链路时延检测系统将是防范DDos攻击最后一道防线，一旦发生链路异常的情况，那么系统可以立即做出反应，争取将损失降到最低。

DDos入侵检测也是网络安全领域的重要研究课题，目的是检测系统或网络中的拒绝服务攻击所带来的安全威胁。一般来说，DDos攻击分为攻击者通过合法的第三方组件隐藏其身份的攻击类型的基于映射的拒绝服务式攻击：包括MSSQL攻击、SSDP攻击、LDAP攻击、NETBIOS攻击等等；攻击者通过占用合法的第三方组件隐藏其身份的攻击类型的基于占用的拒绝服务式攻击：包括UDP攻击、SYN Flood攻击等等；

我们实现DDos入侵检测采用深度学习的相关方法。深度学习在诸多领域应用广泛，如自然语言处理、图像识别、推荐系统等。而且，深度学习技术也展现出针对数据出色的特征提取与处理能力，对于实时流量变化较大的DDos入侵检测，也可以采取当前网络环境和网络流量特性对模型进行针对训练，生成符合特定环境或背景入侵检测模型。

本作品面向链路时延监测和DDos入侵检测这一重要问题，结合软件定义网络(SDN)、带宽占用监测技术、传统入侵检测以及深度学习技术，设计实现了一套基于SDN智能DDos检测防御系统。该SDN智能DDos检测防御系统基于SDN架构，在控制器端应用防火墙策略，指示交换机建立相应流表项，并将经过交换机的报文镜像至DDos入侵检测器。入侵检测器使用深度学习模型针对性训练得到，能够对镜像流量进行分析，提取特征，从中鉴别攻击行为。控制器根据入侵检测器的分析结果自动更新防火墙过滤规则。

本作品拟实现一个易用，可靠，具备轻量级、智能化集成了DDos攻击检测与防御，链路时延和带宽监测的SDN防御系统。其中轻量级指系统能进行实时DDos攻击检测，经交换机的链路时延和带宽监测，并根据检测结果动态更新DDos防火墙规则；智能化指系统能通过学习已有的流量数据集，获得检测未知威胁的能力，而不需依赖特定签名库。

本作品综合应用传统入侵检测、SDN和深度学习技术，研究传统网络功能在SDN架构下的实现，利用深度学习模型潜在的决策优势与SDN易于部署、计算资源集中、能与复杂网络环境互动，动态调整网络策略的优势相结合，探索动态、智能、交互式网络管控的新思路。

## 2 设计思路与方案

本作品实现了一套**基于SDN的智能链路时延监测、DDoS攻击检测与防御系统**。通过基于深度模型的DDos检测器对网络中的流量进行实时监控，基于SDN防火墙响应DDoS入侵检测器来修改路由器流表项，对检测到的恶意DDoS攻击流量进行丢弃或转发，从而实现对DDoS攻击检测与防御于一体，实现智能化、轻量化、自动化的网络流量监控与管理。

我们的系统主要面向SDN网络架构中的DDoS防护问题，为此我们采取入侵检测系统、链路时延检测系统、防火墙以及SDN控制器联动的方式来防御DDoS攻击，入侵检测系统使用AI技术，智能检测网络流量中的恶意流量，将检测结果发送给防火墙。防火墙根据入侵检测系统的告警信息生成防火墙规则，然后SDN控制器根据防火墙规则下发相应流表到交换机，阻断SDN网络与

攻击者之间的流量，实现对DDoS攻击的防御。我们还实现了DDoS防御系统的Web端管理页面，可以查看

网络中的流量状况、增删改查防火墙与入侵检测规则等。

## 2.1 设计思路

我们的系统主要由四大模块构成：SDN带宽占用监测系统、SDN防火墙、DDoS攻击检测器以及管理系统。SDN延时带宽监测系统通过Ryu控制器收到OFP Port Stats Reply消息读取响应信息并计算带宽占用值，通过时间戳算法计算链路时延。SDN防火墙是系统核心模块，通过修改交换机流表项提供基本的网络控制能力，配合交换机应用防火墙策略，将实时流量镜像一份至DDoS攻击检测器供后者分析。DDoS攻击入侵检测器以深度学习模型的形式部署，对捕获的流量进行实时分析，将分析结果传递给SDN防火墙。管理系统主要面向系统中的日志、DDos入侵检测规则以及防火墙规则这三类重要文件，向用户提供日志查看和路由规则编辑界面。

## 2.2 技术路线

本作品使用Python语言，在Linux环境下实现。选用Mininet网络仿真器、Ryu控制器实现防火墙并监测交换机的延时带宽，为系统提供核心防护能力。选用卷积神经网络(CNN)与长短期记忆算法(GRU)训练DDos入侵检测器，为系统提供实时流量分析检测的能力。选用Django和Bootstrap实现管理系统，提供易用实用、简洁明了的web端管理界面。

## 2.3 主要设计方案

在本作品中，SDN防火墙通过南向接口(SBI)协议与交换机进行通信，使用OpenFlow1.3的通信协议。调用ryu的拓扑，向每个交换机发送OFP Port Stats Requests消息请求交换机端口状态，根据收到的回复计算对应端口带宽占用值，发送echorquest标记时间戳，根据收到的时间戳数目计算链路时延。SDN防火墙可以读取防火墙规则文件，根据过滤规则为到来的数据报文构造相应的流表

项，并下发至交换机，指示交换机额外转发一份镜像数据报文到DDos入侵检测器。DDos入侵检测器可以实时保持监听状态，对捕获的数据报文进行检测分析，将检测结果发送至SDN防火墙，后者根据检测结果对防火墙规则进行相应更新。管理系统提供对防火墙规则和入侵检测规则的编辑界面，同时可查看报警信息日志。系统结构如图1所示。

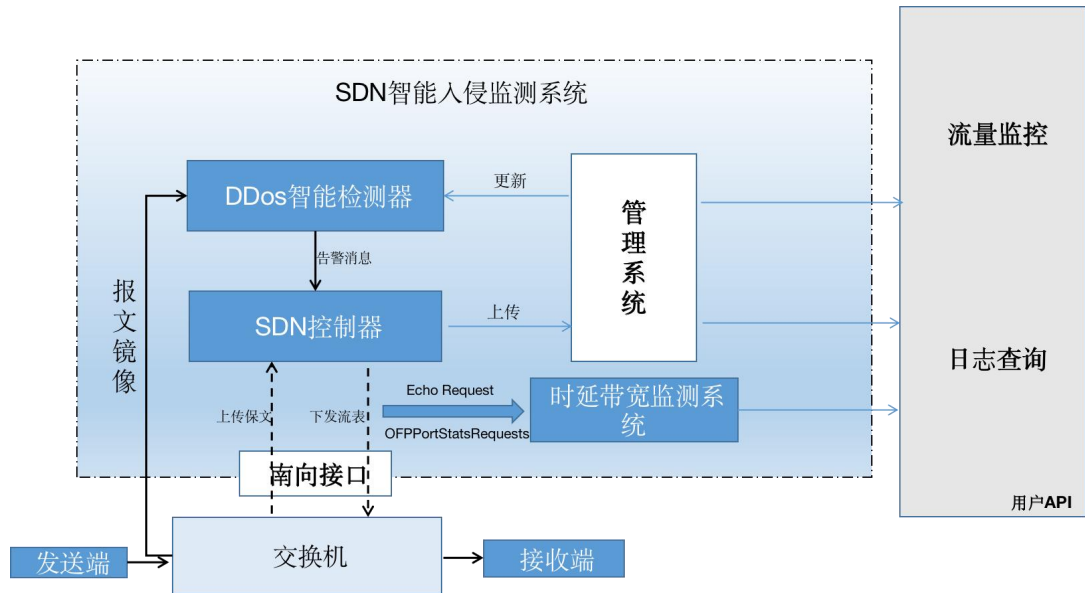


图1 系统结构

下面分别给出SDN防火墙、SDN带宽时延监测、DDos入侵检测器和管理系统的设计方案。

### 2.3.1 SDN时延带宽监测设计方案

Ryu控制器可以调用拓扑API，解析拓扑数据并生成networkx图变量。使用链路带宽占用值作为权值。和每个交换机交换OFP Port Stats Requests请求以及响应，读取Reply消息中数据量统计值，减去上次消息中的统计值，作为对应端口转发带宽占用值，通过交换机id、端口号找到对应链路，记录端口带宽占用值。根据echorequest消息标记时间入口，ryu控制器获取lldp时延，计算交换机之间的时延以及交换机与控制器之间的时延，得到链路时延。

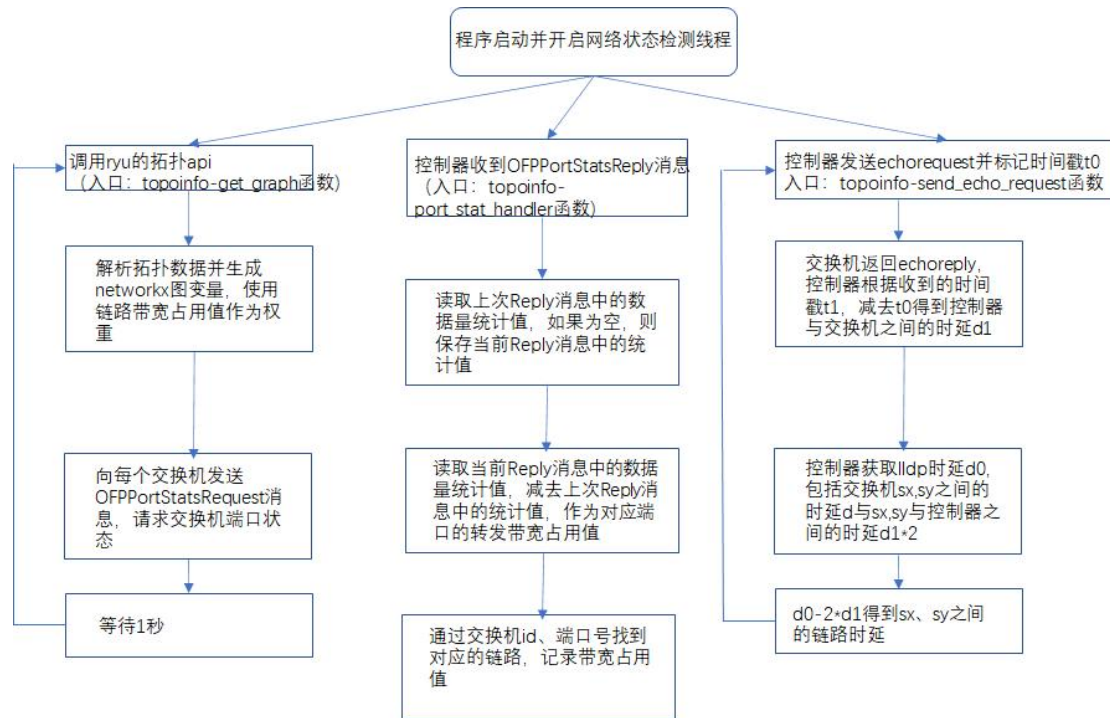


图2 时延带宽监测方案

### 2.3.2 SDN防火墙设计方案

SDN防火墙作为Ryu控制器的应用运行, 可以放行(accept)、重定向(redirect)或丢弃(drop)指定类型的数据报文。SDN防火墙对交换机发来的每个数据报文应用过滤规则, 丢弃该报文或指示交换机进行转发, 同时构造相应的流表项下发给交换机。在流表项中指示交换机将所有匹配报文额外镜像一份至DDos入侵检测器。另一方面, SDN防火墙对入侵检测器进行持续监听, 如果入侵检测器发来报警信息, 就根据检测结果, 查询入侵检测规则, 对防火墙规则进行相应更新, 例如丢弃或重定向来自某个ip或某个端口的流量。防火墙规则与入侵检测规则的具体设计参见第4节。

### 2.3.2 DDos入侵检测器设计方案

本模块采用基于深度学习的IDS技术。基于深度学习的实时IDS技术目前研究较少, 所考虑的主要实现方向为采用Pytorch模块针对性构造深度学习网络模型, 根据现有最新的网络安全数据集DDos2019进行模型训练, 并保存训练模型。在IDS中, 通过针对构造符合数据集的特征提取器, 从实时流量中提取相应特

征，并送入训练完成的模型进行实时流量的分类。因此，选择训练数据集、构造网络模型和实时特征提取成为IDS能否高效的关键。

整体来看，本模块可分为两个部分，即深度网络的训练部分和实时数据包的特征提取与分类部分。

深度网络最终实现的模型主要考虑采用Conv+GRU的形式搭建深度混合神经网络。网络共5层，前1层为卷积层，卷积层后，接一个最大池化层，进行降采样。第3层为GRU层，对形成的抽象特征进行进一步分析。最后两层为全连接层，最终的输出的维度为8。各层后均连接gelu激活函数进行激活。最后将全连接层的输出接入softmax函数进行数据调整，之后输出结果。

实时数据包提取和分类主要通过对实时传入的数据包以及更宏观的连接流进行分析，从而增量的提取连接流的特征并缓存，将缓存的数据进行构造，形成符合深度网络输入要求的数据，并送入训练好的网络进行分类，将得到的结果进行一定的包装返回到SDN防火墙。

### 2.3.3 管理系统设计方案

管理系统基于python框架Django和前端样式库Bootstrap实现，为用户提供日志、防火墙规则和DDos入侵检测规则等重要信息的图形化查询和编辑界面。具体来说，管理系统包括系统首页、防火墙日志、入侵检测日志、管理日志、防火墙规则、入侵检测规则、流量统计、节点状态和系统测试共9个子页面。用户可以通过管理系统查询防火墙日志、入侵检测日志、管理日志和流量统计信息，还可以查看和编辑防火墙规则和入侵检测规则。另外，管理系统还集成了基本的测试功能，用于演示系统的有效性。用户可以通过管理系统指示Mininet中的主机发送数据包，并在流量统计和节点状态等页面查看测试结果。

管理系统的信息交互过程如图所示，后台在Django的视图函数中读取以csv格式存储的日志、防火墙规则和入侵检测规则信息，在预先编写的模板中进行渲染，将得到的html页面传递给前端，前端则将用户对规则文件的编辑操作通过ajax请求传递给后台，后台将请求中携带的数据格式化后写入规则文件，登记相应的管理日志，然后向SDN防火墙发送消息，告诉防火墙规则文件已被用户修改，需要立即应用至交换机的流表项中



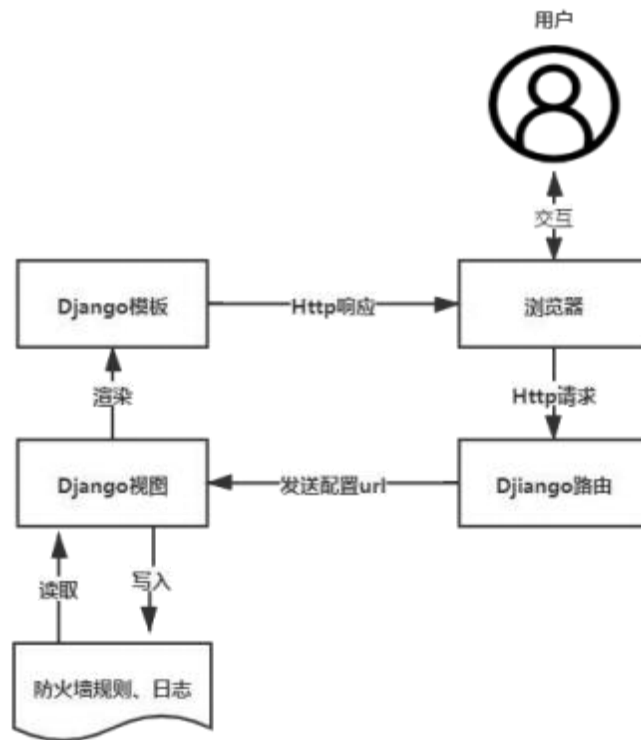


图3 管理系统的交互流程

### 3 方案实现

本作品实现的基于SDN的智能DDoS攻击检测与防御系统包括三个模块：SDN防火墙、DDoS入侵检测器和管理系统。其中SDN防火墙选用Ryu控制器实现，在Mininet网络仿真器上运行，入侵检测器选用Pytorch模块，构建CNN和GRU训练模型得到，管理系统选用Django和Bootstrap实现。下面详细说明各模块的实施细节。

#### 3.1 SDN时延带宽监测实施方案

Ryu控制器通过Openflow协议与交换机进行交互，通过这些协议可以获取交换机的状态信息，从而为实时网络拓扑的流量监测、时延带宽监测提供了基础。本作品采用Openflow1.3协议

### 3.1.1 OFPT\_STATS\_REQUEST&&REPLY消息

OFPT\_STATS\_REQUEST&&REPLY可以获得统计信息，主要用于请求数据或从交换机状态信息我们可以利用统计信息做：负载平衡，流量监控等基于流量的操作。（在1.3中请求被放在一个或多个OFPT\_MULTIPART\_REQUEST消息中，交换机用一个或多个OFPT\_MULTIPART\_REPLY消息进行响应。复合消息用于编码请求或应答那些可能携带大量数据而且不能装进单个OpenFlow消息（仅限于64kb）的情况。）

OFPT\_MULTIPART\_REQUEST消息格式：

version	type	length
xid		
type		flags
body[0]		

OFPT\_MULTIPART\_REPLY消息格式：

version	type	length
xid		
type		flags
body[0]		

OFPMPL\_DESC=0:整体统计信息。请求交换机版本信息，制造商家等信息。

requestbody结构体是空。replaybody回复结构体ofp\_desc\_stats

OFPMPL\_FLOW=1:OFPLST\_DESC单流请求信息， requestbody结构体是：

ofp\_flow\_stats\_request.回复bodyreplay:ofp\_flow\_stats

OFPMPL\_AGGREGATE=2:OFPLST\_FLOW多流请求信息,requestbody结构体是ofp\_aggregate\_stats\_request.回复body结构体ofp\_aggregate\_stats\_replay

OFPMPL\_TABLE=3:OFPLST\_AGGREGATE,流表请求信息： request为空， 回复是流表数组ofp\_table\_stats

FPMP\_PORT\_STATS=4:OFPLST\_TABLE 物理端口信息请求。  
fp\_port\_stats\_request.

ofp\_port\_stats.

### 3.1.2 EchoRequest&EchoReply消息

最常用的ping命令就是使用EchoRequest和EchoReply来实现的。

EchoRequest（回送请求消息）：由源设备（主机、路由器等）向一个指定的目的设备发出的请求。这种消息用来测试目的地是否可达。

EchoReply（回送响应消息）：对EchoRequest的响应。目的设备发送EchoReply来响应收到的EchoRequest。

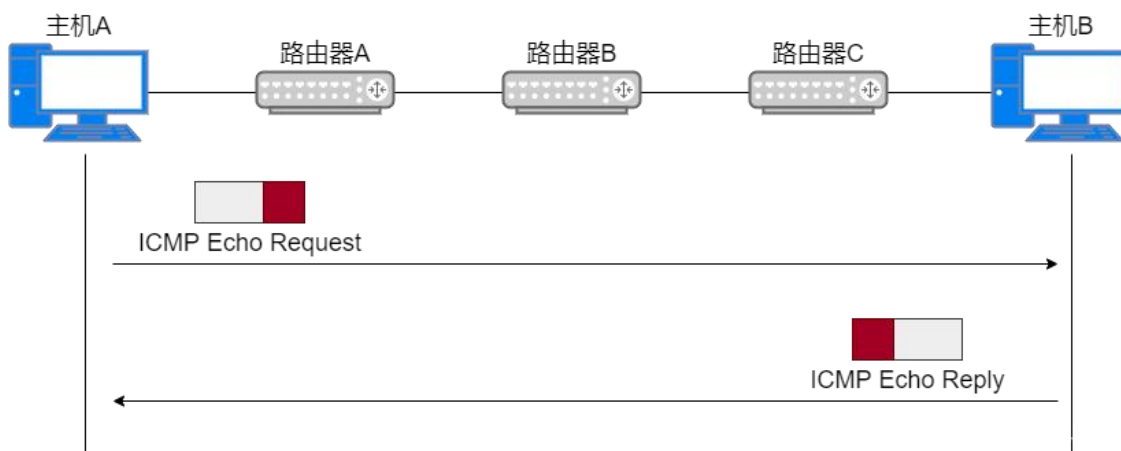


图4

### 3.1.3 回送消息处理与时延计算

接收OFPPortStatsReply消息入口：topinfo-portstathandler函数，读取Reply消息中的数据量统计值，如果为空，则保存到消息栈中。否则减去上次对应的消息数据量统计值，作为对应端口转发带宽占用值。通过交换机的id、端口号找到对应的链路。

echorequest入口：topinfo-send\_echo\_request函数，从echoreply中收到的时间戳t1减去发出request的时间戳t0得到控制器与交换机之间的时延d1。控制器获取lldp时延d0，包括交换机sx,sy之间的时延以及他们与控制器之间的时延d1\*2，d0-2\*d1得到sx,sy之间的链路时延。

## 3.2 SDN防火墙实施方案

SDN防火墙作为Ryu控制器的一个应用，运行在由Mininet仿真出的网络中。

### 3.2.1 Mininet工具

使用Mininet的命令行工具或python接口可以快速创建支持SDN架构的网络，其中包含若干虚拟终端、交换机以及连接它们的链路。网络拓扑由用户自定义，网络中的虚拟终端可以方便地运行宿主机上的程序，这为应用测试提供了良好条件。在Mininet网络初始化时，可以指定控制器地址，从而将网络连接到编写好的控制器应用。在这之后，Mininet中的虚拟交换机与控制器应用之间按照预先选择的SDN南向接口协议进行通信。Mininet支持OpenFlow协议。本作品使用Mininet工具仿真网络测试SDN防火墙以及整个系统的有效性。

### 3.2.2 Ryu控制器

本作品使用Ryu4.34版本。Ryu控制器是一款开源的、基于python的SDN控制器。Ryu控制器良好封装了和交换机之间的通信接口，易于使用。本模块作为Ryu的应用运行。Ryu提供OpenFlow协议中SwitchFeatures、PacketIn等重要事件的处理接口。本模块的主要功能就是通过编写这些事件的处理程序实现。此外，Ryu还提供一套完整的事件机制，允许开发者自定义事件和发送事件到指定实体。本模块通过自定义EventAlert事件，实现了对防火墙规则的动态配置功能。

### 3.2.3 网络拓扑结构

本项目网络拓扑结构主要基于Mininet工具仿真搭建。包括5台主机和一个交换机。具体结构如图5所示。

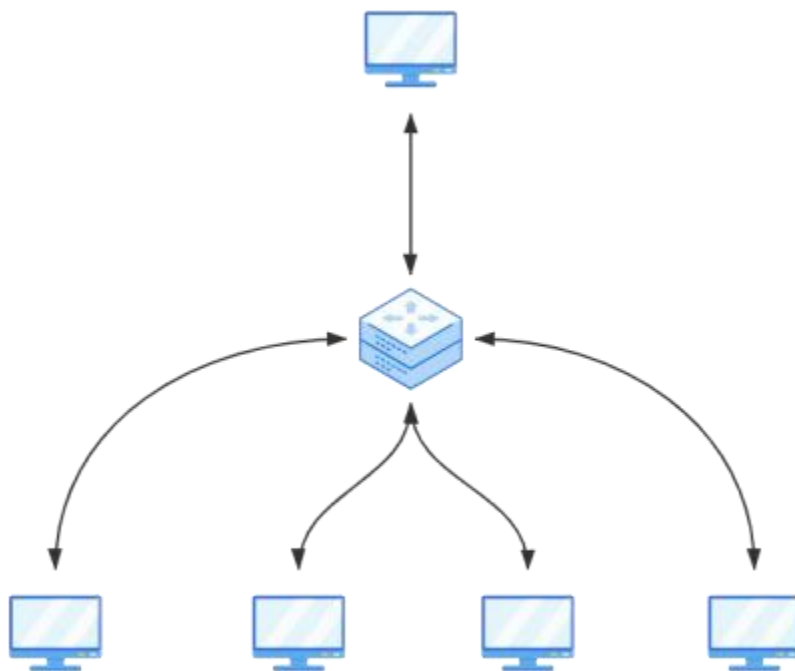


图5 网络拓扑图

### 3.2.4 规则设计

本模块主要包含两类重要规则，即防火墙规则与入侵检测规则。防火墙规则描述对数据报文的响应动作，入侵检测规则描述对DDos入侵检测器检测出的各类型DDos攻击的响应动作。对于防火墙规则，本模块支持放行(accept)、重定向(redirect)和丢弃(drop)三类响应动作。对于DDos入侵检测规则，本模块支持告警(alert)、重定向(redirect)和丢弃(drop)三类响应动作。

一条防火墙规则包括六个字段：id、源ip、源端口、目的ip、目的端口和响应动作，例如：“10005,10.0.0.3,any,10.0.0.5,241,redirect”描述的是，对于从10.0.0.3的任意端口发往10.0.0.5的241端口的流量，应采取重定向动作。该规则用1000这个id进行标识。对于文件中存储的防火墙规则，匹配的优先级从上到下，即匹配最先匹配成功的规则，这样可以为防火墙提供更为灵活的配置能力，例如支持any语法。

一条入侵检测规则包括两个字段：攻击类型和响应动作。其中攻击类型是DDos入侵检测器对数据报文的分析结果。例如：“UDP,drop”描述的是，如果入侵检测器发来消息，指示某个数据报文为UDP攻击,说明该段流量可能存在UDP

类型的DDos攻击，就在防火墙规则中添加一条规则，drop所有从该报文源端口发来，或发往该端口的流量。

#### 3.2.4.1 处理SwitchFeatures事件

SwitchFeatures事件在控制器收到交换机发来的SwitchFeatures消息时触发。交换机通过该消息请求配置，在初始化期间触发。本模块收到该事件后，下发table-miss流表项，即优先级最低的流表项，用于指示交换机在没有其他流表项可匹配时，将数据报文(或其元信息)通过PakcetIn消息发往控制器，由控制器决策转发逻辑。

#### 3.2.4.2 处理PacketIn事件

PakcetIn事件在控制器收到交换机发来的PakcetIn消息时触发。本模块通过处理该事件实现防火墙的核心防护功能。本模块收到该事件后，读取防火墙规则文件，逐条与数据报文匹配，匹配成功后根据规则中指出的响应动作决定将该数据报文正常转发/向重定向端口转发/丢弃，其中重定向端口已经预先配置。

在遍历规则的同时，为每条规则构造相应的流表项，下发至交换机，用于交换机匹配之后到来的数据报文。交换机根据这些流表项，将相应的数据报文正常转发/向重定向端口转发/丢弃。这些流表项还指示交换机，对于没有丢弃的报文，应额外转发一份至入侵检测器监听的端口。该端口同样已经预先配置。另外，参照Ryu官方提供的示例代码，在该事件中实现了简单的mac地址学习功能。

#### 3.2.4.3 处理EventAlert事件

EventAlert事件是本模块的自定义事件，在监听到DDos入侵检测器发来的告警信息时触发。该事件本身基于Ryu提供的事件机制实现。即通过send\_event接口发送事件，通过@set\_ev\_cls修饰器指示对应事件的处理程序。本模块通过处理该事件实现对防火墙规则的动态配置。本模块收到该事件后，从该事件附带的告警信息中提取数据报文的源ip、源端口以及入侵检测器认定的攻击类型，然后读取入侵检测规则文件，找到攻击类型对应的响应动作，构造相应的防火

墙规则，更新至防火墙规则文件中。由于防火墙规则按从上至下的优先级匹配，因此新规则插入到规则文件头部。

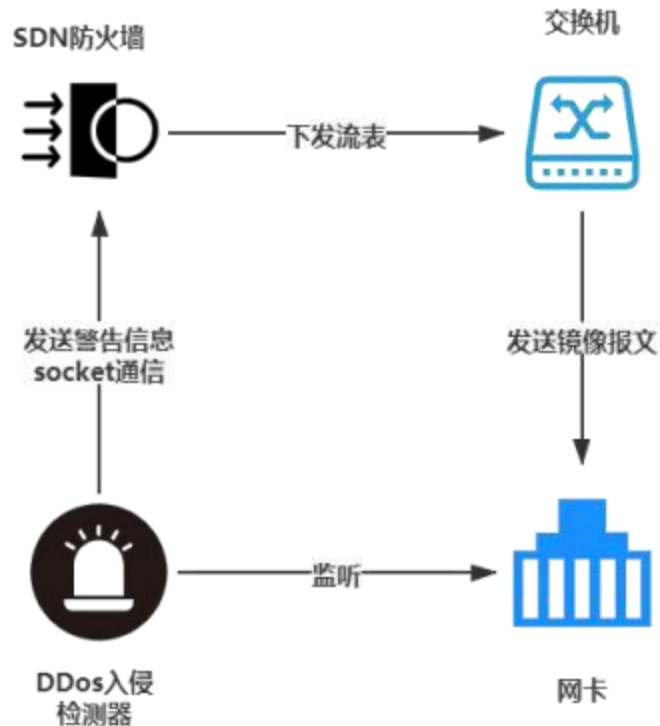


图6 SDN防火墙与DDos入侵检测器的联动

### 3.2.5 连接DDos入侵检测器

本模块与DDos入侵检测器的连接包括两部分实现。第一，使用scapy库对指定网卡进行监听，将监听到的数据包作为参数传递给被调用的入侵检测器，并取得入侵检测器返回的检测结果。第二，封装告警信息，在其中包含检测结果、被检测数据报文的源ip、源端口、目的ip、目的端口以及原始报文数据。将告警信息通过socket发送给SDN防火墙。防火墙收到后触发EventAlert事件，将告警信息传递给EventAlert处理程序。检测过程中，交换机、SDN防火墙和入侵检测器的关系如图6所示。

## 3.3 DDos入侵检测器实施方案

DDos入侵检测器主要包括两个部分，即深度网络的构建及训练部分和实时数据包的特征提取与分类部分。其中深度网络的有效及针对训练是其功能能否高效率、高准确性实现的基础。下面分别详细介绍两部分的实现方案。

### 3.3.1 深度网络的构建及训练

在深度学习项目及深度网络构建中，我们考虑了以下几个因素：数据集的选取、数据特征的选取、网络结构的搭建、激活函数的选择、网络超参数的选择等。在本次实验过程中，也是如此。由于数据集的选取合特征的选择在前文已经叙述，故在本节不再详细说明，数据集我们对比之后选择了CIC-DDoS2019，这是最新的针对DDos攻击的数据集，它弥补了前面相关数据集的不足，并提出了一种新的基于一组网络流特征的检测和族分类方法来生成的数据。

在其中的80个特征中，我们挑选了其中的25个便于区分不同DDos攻击及正常流量的特征用于构建模型，标签包含7类DDos攻击流量和正常流量。各类标签的含义如下：

Benign类，正常流量。

Portmap类，DDos攻击流量，Portmap是一种对TCP或UDP111端口的攻击(111端口是一种用于将客户端定向到正确的端口号的服务，以便客户端可以与请求的远程过程调用(RPC)服务进行通信)。

LDAP类，DDos攻击流量，LDAP注入是一种攻击，用于利用基于用户输入构造LDAP语句的基于web的应用程序。

MSSQL类，DDos攻击流量，微软结构化查询语言，(MSSQL)注入是一种攻击，它使执行恶意SQL语句成为可能。

NETBIOS类，DDos攻击流量，针对网络基本输入/输出系统(NetBIOS)中的安全漏洞，这些漏洞允许攻击者通过网络查看计算机内存中的信息。

SYN类，DDos攻击流量，SYNflood是一种拒绝服务攻击，攻击者向目标系统发送连续的SYN请求，试图消耗服务器资源，使系统对合法流量没有响应。

UDP类，DDos攻击流量，用户数据报协议(User Datagram Protocol,UDP)泛洪攻击是一种将大量UDP数据包发送给受害者，使其无法处理和响应的攻击。因此，耗尽保护目标服务器的防火墙。



UDPLag类，DDos攻击流量，UDPLag是一种破坏客户端和服务端之间连接的攻击。

### 3.3.1.1 DDos攻击数据集的选取

在DDos数据集的选择中，有若干可以进行训练的网络安全数据集，主流包括：DDoS2019、DoSdataset(2017)、DDoSAttack2007等。其中数据集DDoS2019弥补了前面相关DDos数据集的缺点并进行了改进，是建立在实时流量上提取的最新的数据集，故在此次实验中我们采用了2019年提出的DDoS2019数据集进行模型的训练。

在原始数据集中主要包括13个类别的DDos攻击流量信息，包括NTP、UDP、DNS、LDAP、MSSQL、NetBIOS、SNMP、SSDP、SYN、UDP-Lag、Web-DDoS、TFTP和Portmap，和一种正常的状态的流量信息Benign。数据集中共有80个关于实时流量的特征，包括流量的基础特征、网络的统计状态信息等。在此次实验中，我们使用的.csv数据集主要包含7类DDos攻击。对于攻击的分布如下，Portmap类共有19万条数据，LDAP类约有201万条数据，MSSQL类约有582万条数据，NETBIOS类约有346万条，SYN类约有458万条，UDP类约有437万条数据，UDPLag约有8万条数据，各类攻击的分布如图7所示。

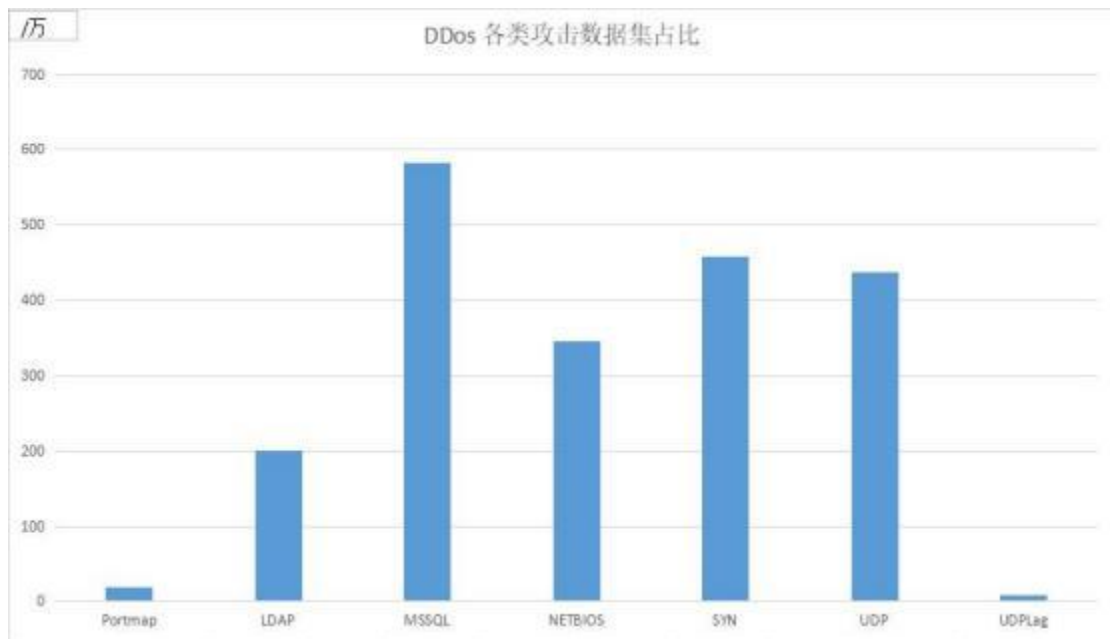


图7 DDoS2019初步数据集分布

由于各类DDos攻击的数据集初步分布不均匀，直接对这样的数据集(类型分布不均匀)进行训练会影响到模型分类的准确度，所以我们需要对数据集做过采样或欠采样处理，实验过程中，我们采用欠采样对数据集进行初步处理，过程如下：

以数据集数量最少的UDPLag的数量为基准，我们在其他6类DDos攻击中各自均匀不重复的抽取了约8万条数据，以达到数据集各类数据分布的均匀性，并将每类数据打乱，避免抽取顺序对数据集造成的影响。经过处理后的DDos攻击数据集分布如图8所示。

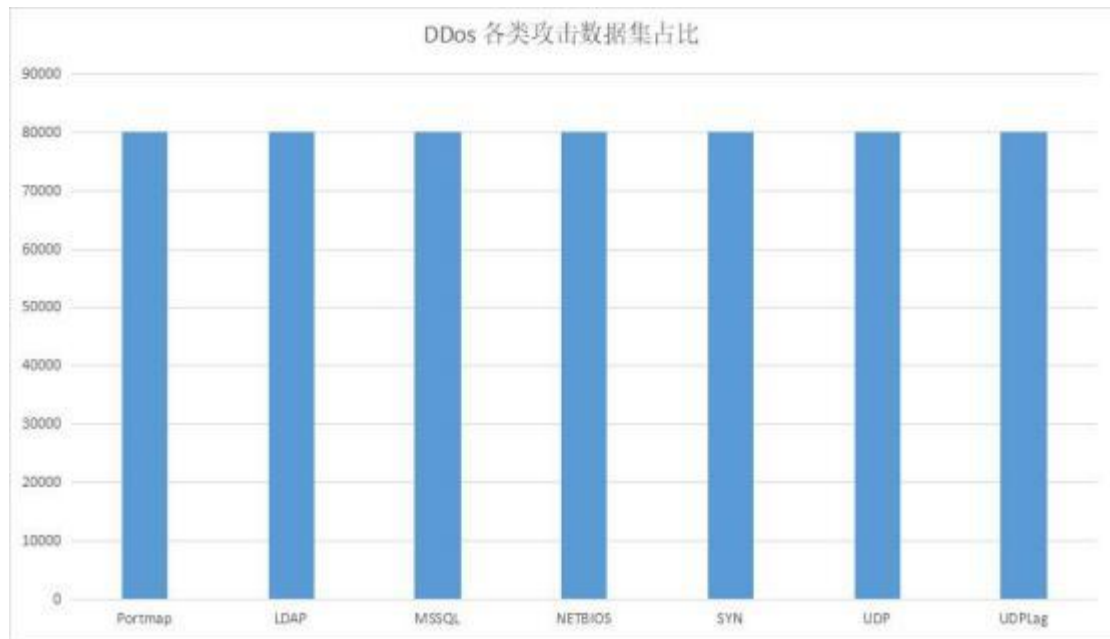


图8 DDos2019训练用数据集分布

对于DDos2019数据集中原有的80个特征，考虑到各类特征的针对性，其中部分特征对于普通流量与DDos攻击流量或DDos攻击流量之间的区分影响不大，以及各类特征针对的不同DDos攻击，由于不同的特征对不同的DDos攻击影响程度也不同，我们从80个特征中挑选了约30条针对不同DDos攻击具有显著区分程度的特征。

考虑到本次搭建的IDS与模拟SDN环境的兼容情况，我们酌情删去数据集中若干特征进行训练，最后选取了其中25条流量特征。虽然这样可能会影响模型的准确率，但却可以更好的与IDS形成联动，以及更好的符合我们所搭建的网络环境。其中挑选的特征如下表一：

表1模型训练考虑的流量特征

Feature	Description
Fwd Packet Length Max	Maximum packet size in the forward (outgoing) direction
Fwd Packet Length Min	Minimum packet size in the forward direction
Min Packet Length	Minimum length of a packet
Max Packet Length	Maximum length of a packet
Average Packet Size	Average size of a packet
FWD Packets/s	Number of forward packets per second
Fwd Header Length	Header length of a forwarded packet
Fwd Header Length 1	Number of bytes in a header in the forward direction
Min_Seg_Size_Forward	Minimum segment size in the forward direction
Total Length of Fwd Packet	Packet size in the forward direction
Fwd Packet Length Std	Standard deviation of a packet in the forward direction
Flow IAT Min	Minimum time between two packets in the flow
Subflow Fwd Bytes	Average number of bytes in a sub flow in the forward direction
Destination Port	Address to receive TCP or UDP packets
Protocol	TCP or UDP for data transmission
Packet Length Std	Standard deviation of the packet length
Flow Duration	Duration of the flow in $\mu$ s
Fwd IAT Total	Total time between two packets in the forward direction
ACK Flag Count	Number of packets with ACK
Init_Win_Bytes_Forward	Number of bytes in initial window in the forward direction
Flow IAT Mean	Mean time between two packets in the flow
Flow IAT Max	Maximum time between two packets in the flow
Fwd IAT Mean	Mean time between two packets in the forward direction
Fwd IAT Max	Maximum time between two packets in the forward direction

在完成模型的训练后，根据SDN防火墙可以提供的信息构造实时IDS。实时IDS所接受为网络的原始流量，通过程序提取所需的特征，并汇总特征送入深度网络中进行分类。由于网络流量的实时采集较为困难，还原DDos2019数据集的网络环境与监视情况同样较为困难，故考虑采用对同一流量中的各个数据包进行增量预测，即从发起建立连接的第一个数据包到最后一个数据包增量的记录流量的统计特征进行预测，如果连接进行到任意程度时IDS报警，立即通知SDN防火墙，终止连接。考虑的主要特征如表1所示。

### 3.3.1.2 深度网络的基准搭建

在初始阶段，考虑首先在经典的传统深度网络结构模型Le-Net5基础上进行搭建。传统的Le-Net5网络模型如下图5所示。但LeNet5网络应用于2维的图像卷积，与本项目的应用有所差别，本项目由于需要对一维的数据提取相应特征，所以主要采用1维卷积进行实现，故在初始的阶段需要进行一定的调整。

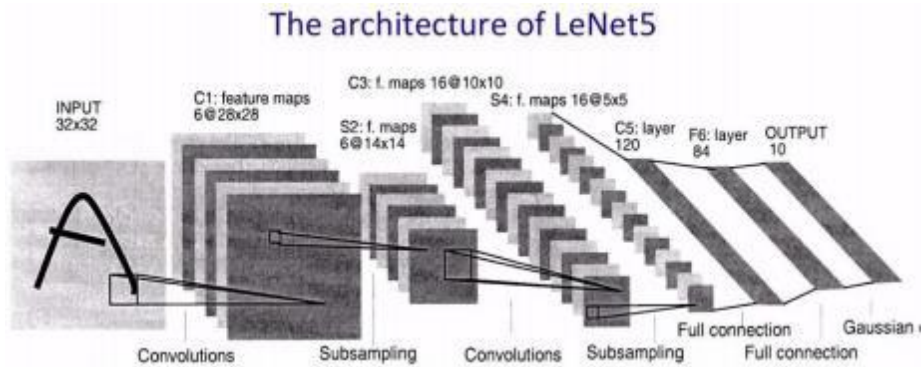


图9 Le-Net5网络结构

由于所选用的特征共30种，经过预处理后输入网络的特征共25个，故在网络的层数选择上，不能完全按照Le-Net5的结构及卷积核参数选取来搭建卷积，因为这会导致最终在最后一层卷积处输出的特征过少，无法通过全连接层得到有用的权重分配，故网络结构以及数据的输入情况要相应进行调整，在本项目中考虑不将数据扩展为2维类似图片的形式，仅采用1维数据，即输入的单条数据为1\*25的维度，并通过1D卷积进行处理。

由此在初始阶段，考虑了两种不同复杂度的网络进行基准的网络搭建，如下图6，7所示。网络接收的输入数据维度为1\*25，网络结构-1采用2个卷积层，1个max-pooling层和2个全连接层。其中卷积层的卷积核为1\*3，max-pooling的卷积核为1\*2，最终输出维度为1\*8。网络结构-2采用了4个卷积层，2个max-pooling层和2个全连接层，其中在每两个卷积层中间接一个max-pooling层，其余网络的卷积参数与网络结构1相同。

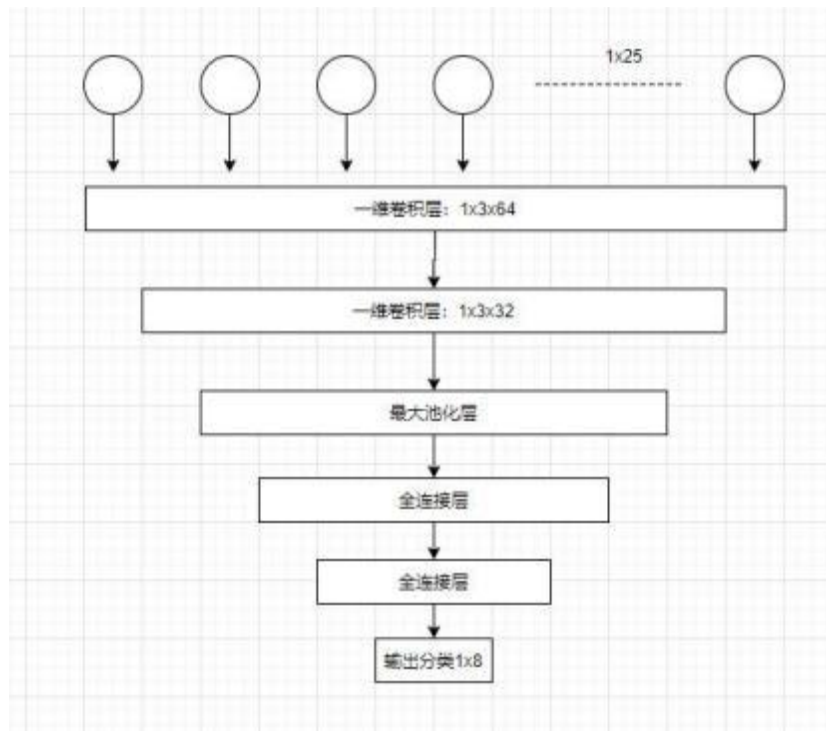


图10 网络结构-1

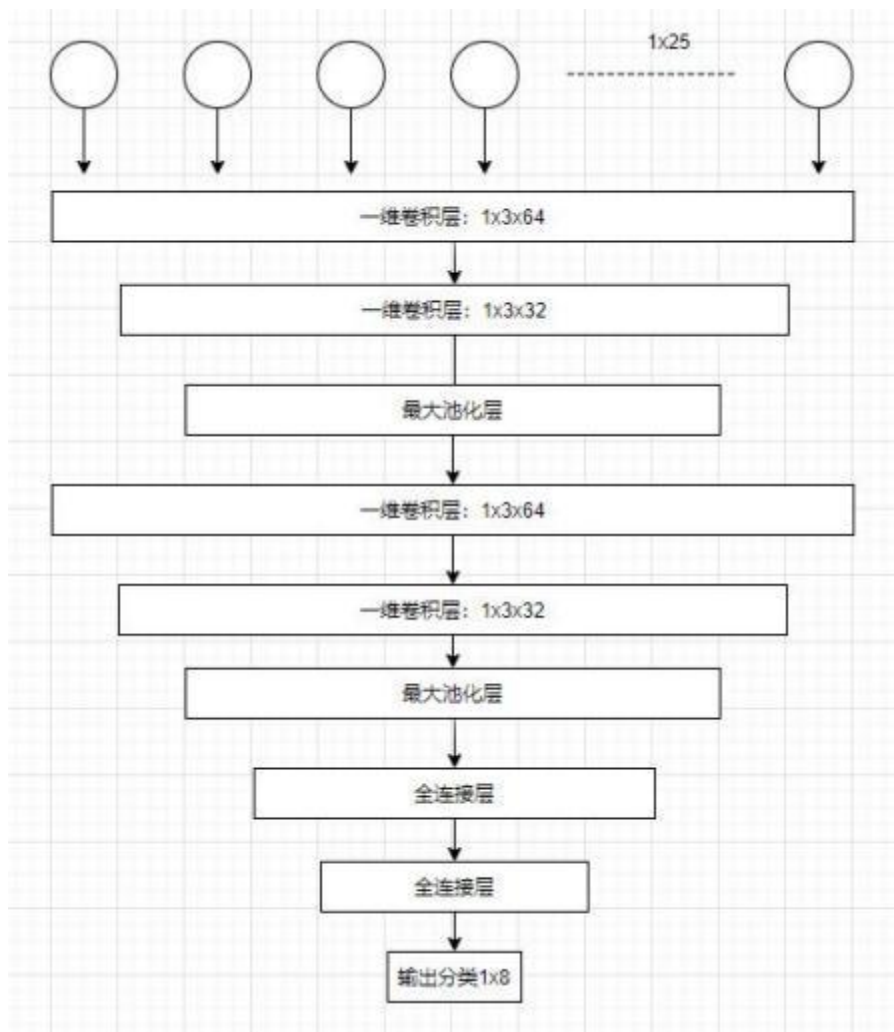


图11 网络结构-2



在测试时，采用迭代次数为250次，应用Adam(自适应梯度下降)进行优化，损失函数采用交叉熵损失函数，学习率为0.00001，权重衰减为0.0001。但在初步的实验训练及测试中，我们发现此模型的效果并不理想，如图12，13所示。由图片中的测试结果可以看出两种模型的效果并不好，对于多分类平均准确率仅分别有不到50%，初步考虑可能是由于数据连续性，仅通过简单卷积网络难以实现对DDos这样连续性攻击数据流特征的完全函数拟合。两个网络的准确率均较差，但网络结构-1的学习稳定性较好，而网络结构-2的稳定性较差，迭代中出现过多次准确率的较大反复，考虑原因为数据的特征较少，在面对多层卷积网络，尤其存在较多max-pooling时，数据维度被迅速压缩，使得有效信息变得更加模糊，从而导致训练中准确率出现较大丢失；同时网络结构过于简单，无法提取出更加细致的特征，使得多分类的准确率仅在50%的左侧，故考虑对网络进行进一步改进。

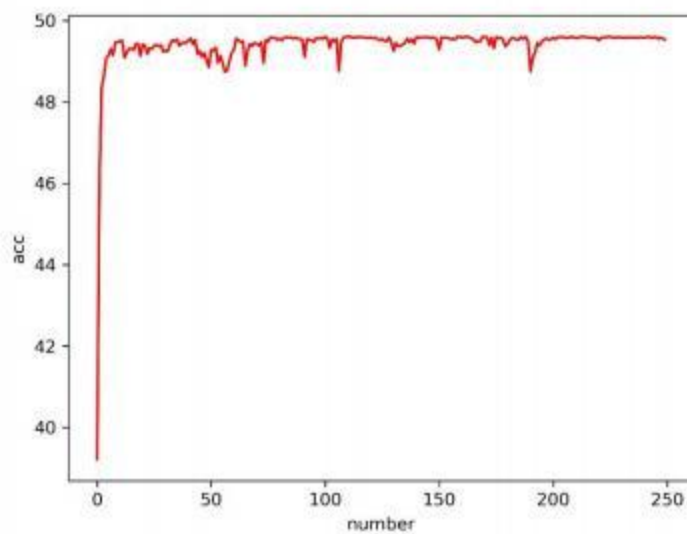


图12 网络结构-1在测试集上的效果图

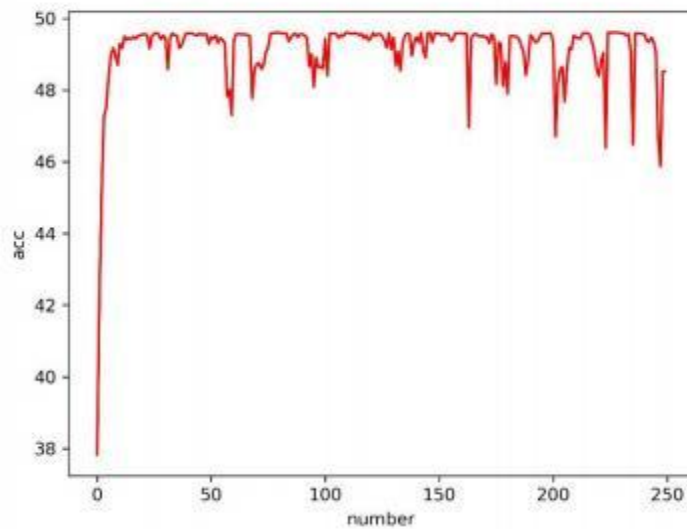


图13 网络结构-2在测试集上的效果图

### 3.3.1.3 引入循环神经网络模型(LSTM和GRU)

经过基准的纯卷积网络的搭建，我们发现仅仅依靠卷积网络对数据进行特征提取无法得到较为良好的效果，考虑到DDos攻击具有连续性和持久性，我们在模型中引入循环神经网络模型。这种循环时间模型可以更好的提取连续性数据的相关特征，具有代表性的如LSTM和以及它的变体GRU等深度神经网络。

LSTM又称作长短期记忆网络，是一种更加特殊的RNN网络，其网络的设计初衷为的是解决长期依赖问题。传统的RNN网络的抽象结构如下图14所示。

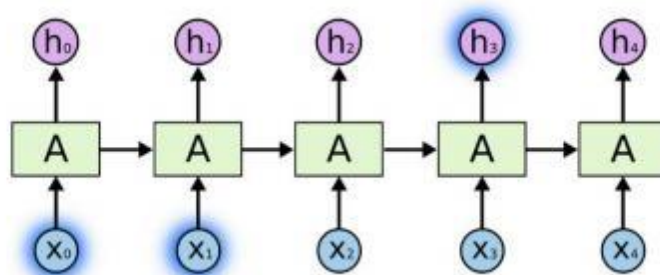


图14 传统RNN网络的抽象结构

当若干RNN单元串联用于解决复杂的任务，如长句子、文章或较多连续时间片的预测任务时，由于RNN缺乏存储能力，同时有用信息与需要进行处理信息的地方之间的距离较远，这样容易导致RNNs不能学习到有用的信息，最终推导的任务可能失败。如下图15所示。

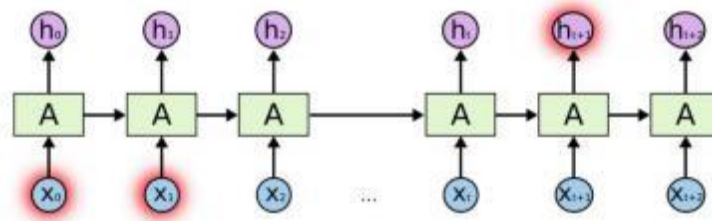


图15 RNN预测失败

由此引入了具有更强记忆性和选择性的LSTM，LSTM单元的结构如下图16所示。由于LSTM内部采用包括忘记门、输入门和输出门的抽象门结构，所以在学习中LSTM单元可以决定需要丢弃哪些信息，添加哪些新的信息。这样使得采用LSTM单元构建的循环网络，在用于解决DDos这样连续性攻击数据流的特征提取以及分类问题时，可以更好的对特征数据以及数据流之间进行联系与分析。

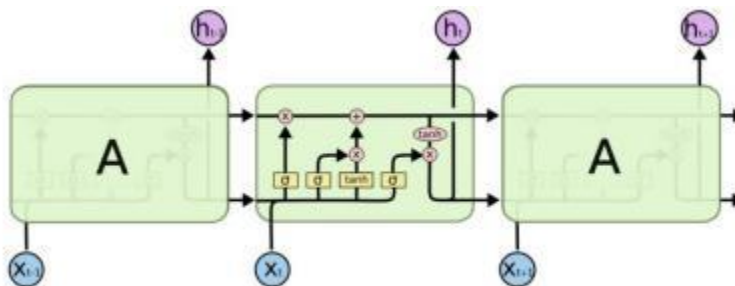


图16 LSTM的结构示意图

虽然LSTM和传统RNN对比下拥有较大优势，但LSTM单元的内部结构较复杂，参数众多，对于在复杂场景下训练效率较差，耗时较长，故产生了一种更加轻量级，但是效果与LSTM相似的结构，即门控循环单元(GRU)，如下图17所示。

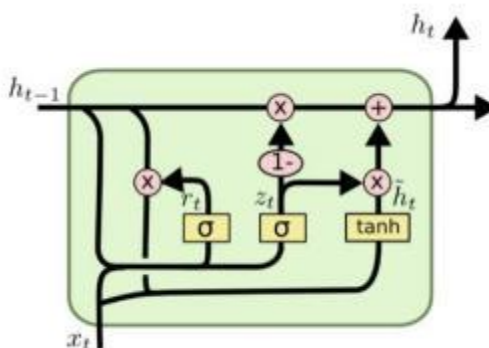


图17 GRU的结构示意图



GRU采用的思想和LSTM类似，但在内部结构中有所变化。GRU将LSTM的忘记门和输入门合并成了一个新的门控结构，称作更新门，定义了前面记忆保存到当前时间步的量。此外GRU还有一个重置门，决定如何将新的输入信息与前面的记忆相结合。

在本项目中，考虑采用CNN和LSTM/GRU的混合神经网络的形式用于解决DDos入侵检测的问题。

#### 3.3.1.4 完整的深度学习网络搭建

经过若干对比实验，我们最终采用的网络模型结构大致如图18所示。

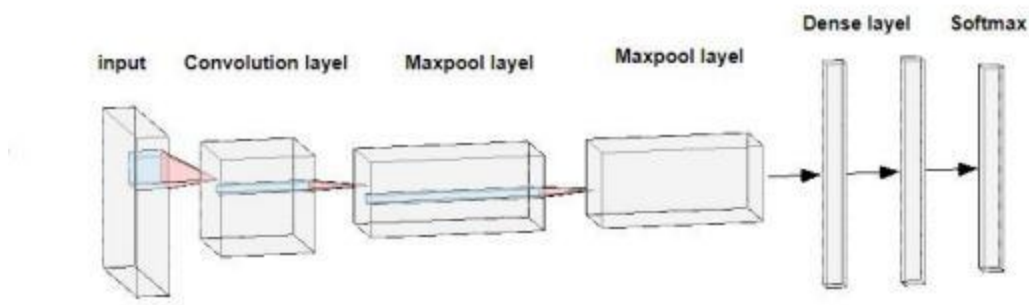


图18 网络结构

为了评估不同网络对此任务的效果，故采用4种不同的网络用于比较，具体配置如下：由于数据是1维的，故Conv层采用1-D卷积进行特征的提取。前2层为卷积层，卷积核大小为3。两个卷积层后，接一个kernel大小为2的最大池化层，进行降采样。第3层为LSTM或GRU，其隐藏层大小为48。最后两层为全连接层，最终的输出的维度为8。各层后均连接gelu激活函数进行激活。最后将全连接层的输出接入softmax函数进行数据调整，之后输出结果，如图19、20混合网络所示，即2层卷积层加1层LSTM/GRU层的混合网络结构。

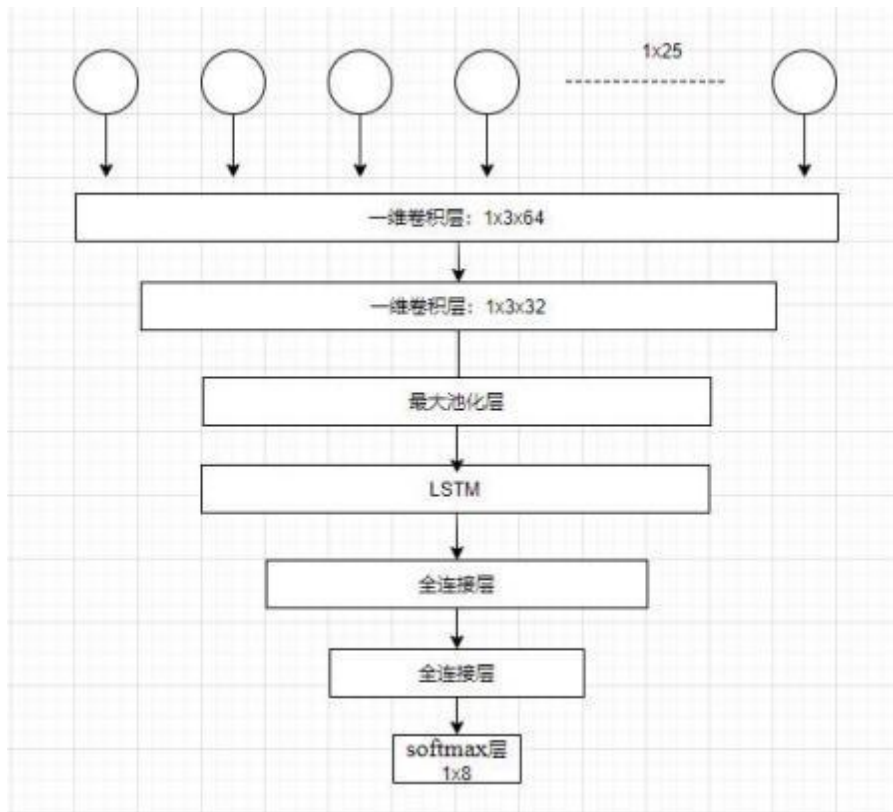


图19 混合结构-1

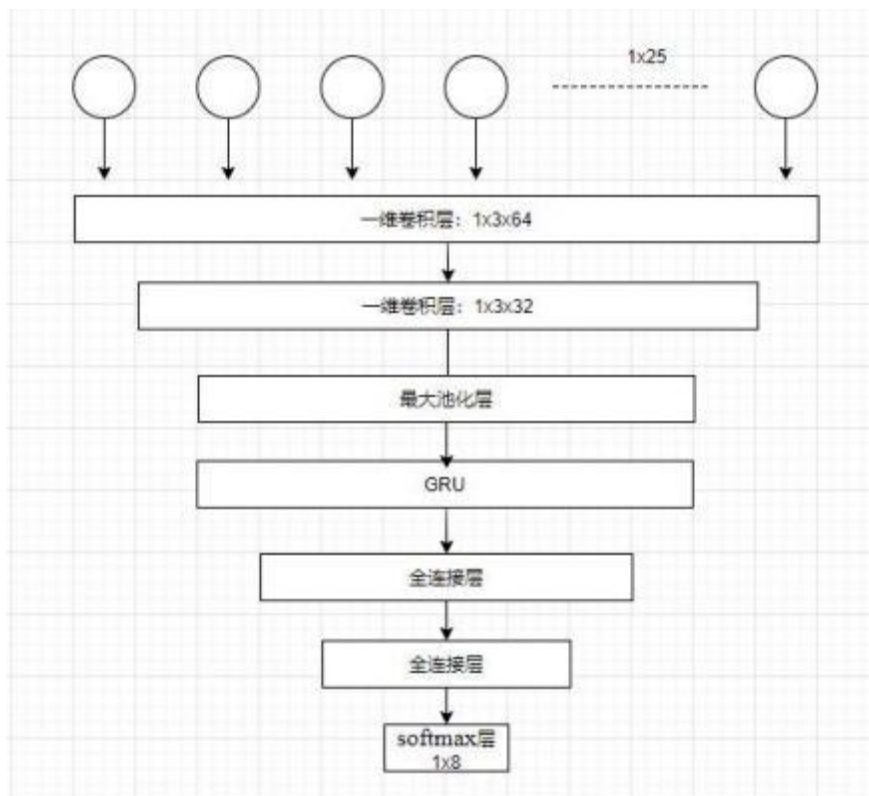


图20 混合结构-2

经过对上述4种不同构造的网络进行对比试验，实验采用250次迭代，采用Adam算法用于优化函数，损失函数采用交叉熵函数，结果如下。

首先是6层网络的结果对比，即2层卷积层，1层下采样层，1层LSTM/GRU层，2层全连接层，如图21，22所示，其中图21为LSTM的混合网络，图22为GRU的混合网络。

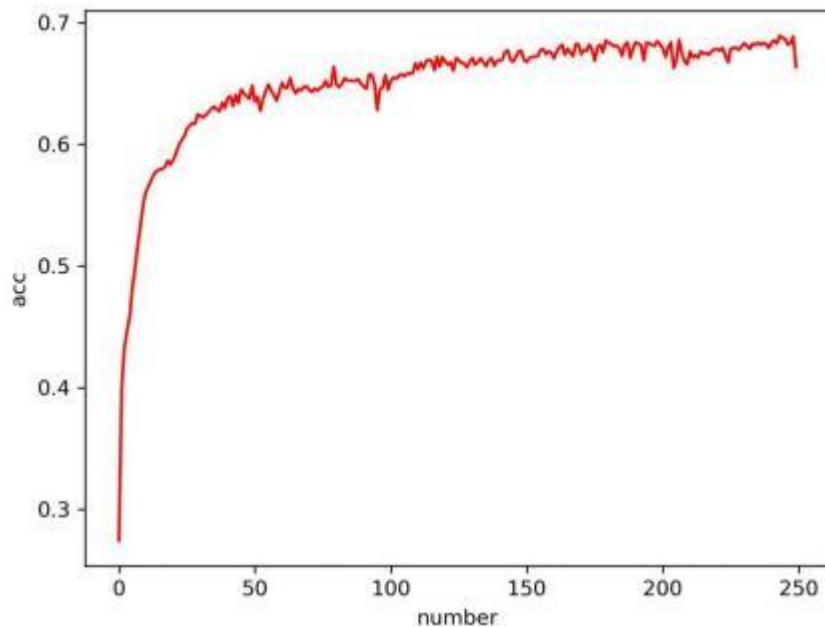


图21 7层GRU混合网络多分类测试准确率

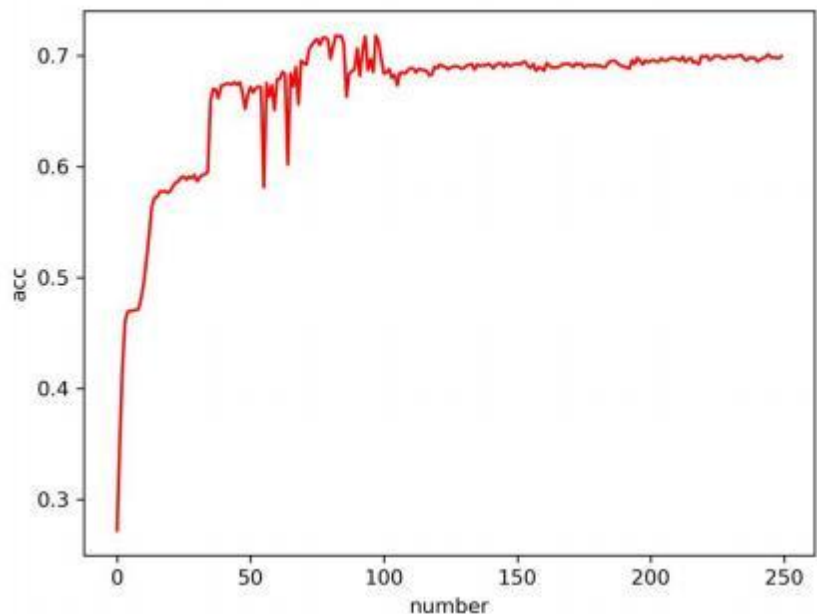


图22 7层LSTM混合网络多分类测试准确率

经过实验结果的对比，可以看出相对于纯卷积网络的<50%的准确率，混合网络的准确率接近70%，性能提升显著，对于多分类，效果提升了超过20%，

模型效果良好。另外，通过对比上述两种采用不同循环网络的混合网络结果，可以发现采用LSTM的模型准确率稍高于GRU，这是因为LSTM的内部逻辑要比GRU更加复杂，对于特征的提取和理解要更优。同时，LSTM的混合网络的准确率波动较大，而GRU虽有准确率波动但是整体上，模型效果更稳定，这样的结果也是缘于LSTM的内部复杂性，使得在不同次训练迭代后，LSTM单元可能会有权重上更多的调整与适应。

由于特征之间关联性并不是特别强，更多的是单个特征对于不同DDos攻击产生的影响有所区别，过多的卷积层反而会影响到单个数据的区分度，所以我们在之后的对比实验中适当减少了模型中卷积层的数量，也适当减小了迭代次数。

下面是减少卷积层后模型5层网络的结果对比，即1层卷积层，1层下采样层，1层LSTM/GRU层，2层全连接层，如图23，24所示，其中图23为LSTM的混合网络，图24为GRU的混合网络。

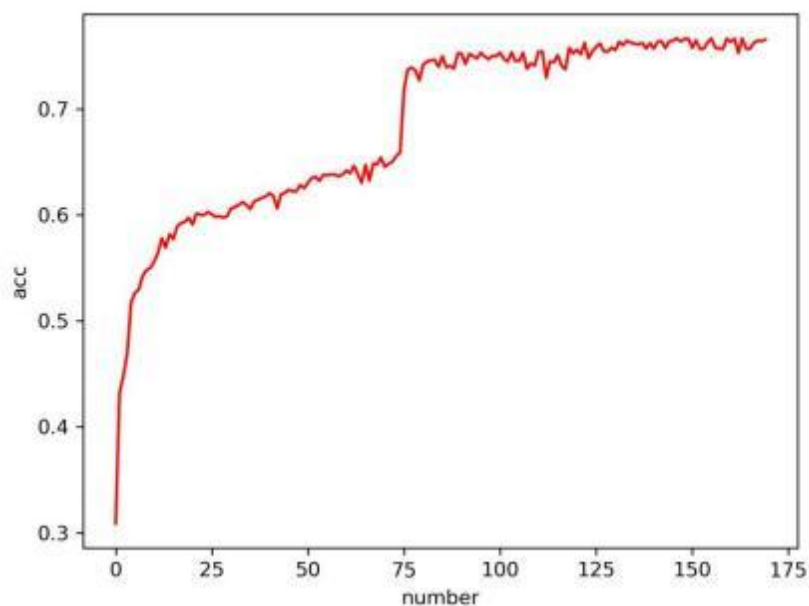


图23 LSTM混合网络多分类测试准确率

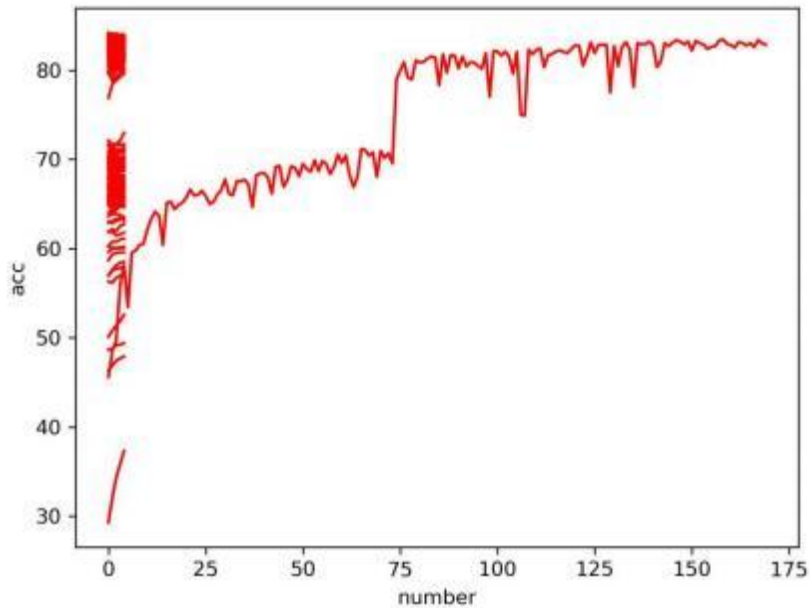


图24 GRU混合网络多分类测试准确率

通过对比上述两种采用不同循环网络的混合网络结果，可以发现采用GRU的模型准确率显著高于LSTM，达到了84.2%，此准确率也是实验中选取的6种主要模型中，准确率最高的，效果最好的模型。1层卷积从原始数据中提取出具有代表性与关键性的抽象特征，经过GRU的分析处理，使得模型的整体效果产生了较大的提升。可以看出，GRU的稳定性和准确率也是在卷积加成下得到了进一步稳固和提升。

经过对比和尝试，最终在IDS里选择的深度学习模型为5层的CNN-GRU混合模型。

### 3.3.1.5 模型搭建过程的补充说明

**激活函数的选取。**抛弃了以往常用的sigmoid, tanh以及relu等，而采取了更加新颖的gelu激活函数进行激活。激活函数的作用主要是为神经网络引入非线性，可以使神经网络的表达能力更加强大。Sigmoid激活函数会使网络遭遇梯度消失或梯度爆炸问题，relu虽然在深度学习中应用广泛，处理高效，但不能避免梯度爆炸，同时可能会遭遇网络大部分分量为0的情况。而gelu函数作为更加新颖的一个激活函数，在本模型的实验中取得了比relu和tanh更加优秀的效果，故在最终的模型中我们选择gelu作为最终的激活函数。

**优化函数和超参数的选取。**主要采用torch模块完成深度网络的构建与训练过程。优化函数采用Adam(自适应梯度下降算法)，损失函数采用交叉熵损失函数。此外，学习速率和权重衰减最终分别选定为 $1e-5$ 和 $1e-4$ ，这种超参数的选取是通过多次实验对比，从而确定的，在此参数下模型的效果相对最好。此外，在现象层加入权重衰减是必要的，因为数据特征以及数据集相对而且数量上并不充分，所以需要一定的正则，避免模型应参数过多而训练集较小出现过度的过拟合现象。

**模型的训练。**由于模型较为复杂，较多次数的迭代会导致过拟合的发生，故为了降低过拟合的情况，在全连接层后增加dropout的过程，在第一个全连接层以25%的概率进行剪枝，在第二个全连接层以50%的概率进行剪枝。因为第一个全连接层需要直接从LSTM中获取输出的特征信息，故过多的dropout会影响输出的结果，所以采用此种方式进行。源数据集共30个特征，根据在比赛中所采用的SDN模拟网络环境情况与数据流量情况，经过提取，保留其中25个主要特征进行分类，精度有所下降。

**深度网络训练使用的函数。**在此模块中主要有4个函数或类型，即load\_data()、data\_assign()、train()和test()。各个函数的实现的功能与整体的执行流程如表2所示：

表2 深度网络训练的主要函数

相关函数	作用
load-data(root)	用于从.csv文件中读取训练集或测试集数据，并对数据进行初步预处理，分开特征与标签。
Train(model,feature_train,label_train,optimizer,criterion,epoch,Device)	用于网络模型的训练。
data_assign(feature,label)	用于对load_data函数返回的特征数据与标签数据进行处理。
Test(model,feature_train,label_train,criterion,epoch,Device)	用于网络模型的测试。

**深度网络训练程序的主流程。**通过load\_data()读取训练和测试集数据，通过data\_assign进行数据类型转换，实例化深度神经网络，初始化训练所需基本参数，通过train()和test()迭代进行模型训练和测试，并根据测试准确度保存模型。

### 3.3.2 实时数据包的特征提取与分类

IDS(intrusion detectionsystem)入侵检测系统是一种对网络传输进行即时监视，在发现可疑传输时发出警报或者采取主动反应措施的网络安全系统。本项目所实现的是一个利用深度学习模型对网络流量进行检测的IDS，可以认为本项目的IDS为一种基于深度学习异常检测技术的入侵检测系统。当本项目的IDS根据分析判断出连接存在异常时，会和SDN防火墙进行联动，从而对连接进行处理，同时更新SDN的内部流表项。

#### 3.3.2.1 IDS入侵检测器的特征提取

由于IDS中的主要分析模块由深度学习的模型提供，所以如何从复杂抽象的网络连接数据中提取和训练集高度符合的特征，成为了IDS能否正常高效实现的重要环节。在实现中，我们考虑根据DDos2019数据中被我们采纳的特征含义，通过对数据包内容分析和连接流的统计分析提取出符合的特征，组成特征块运用深度学习模型进行分类分析，输出IDS的结果。

实时数据包的特征提取与分类的具体实现。关于数据包内容分析和连接的统计信息主要通过python的scapy和dpkt模块进行提取。IDS接收从SDN传递的原始流量数据包，根据源、目的ip和源、目的端口对连接进行分割和判断。同时，根据DDos2019数据集的相关特征的提取方法，相应的从数据流中提取特征并保存，部分特征需要根据一定的算法，从已有的特征中生成。之后，生成用于分类的数据组，送入深度神经网络模型进行分类。深度神经网络采用上文讨论过的CNN+GRU的混合网络实现，在IDS中提前加载已经训练好的网络参数，并实例化网络嵌入IDS模块。由于单个乃至少量数据包所提供的信息过少，我们考虑当同一个连接的数据包不断传递到IDS时，IDS会根据接收到的数据包情况进行增量的更新连接的统计特征，同时送入深度CNN-GRU混合网络进行分类。当此连接完成后，IDS根据流量的特征(ip和端口号)中断记录此连接的特征，同时释放对应缓存的数据。

### 3.3.2.2 IDS入侵检测器的流程

实时数据包的特征提取与分类的主流程。通过`get_input()`读取数据，初步分析数据包，决定是否丢弃。同时，在`get_input()`中，通过`find_seq_id()`函数进行连接流的匹配，以判断是新流还是已经存在的连接流，对于新连入的连接流，采用`init_pool()`函数进行流特征和统计特征的初始化。在对初步的连接处理和ip、端口等特征提取后，使用`preprocess()`函数对所需的特征进行广泛的提取并缓存。针对我们所采用的增量处理办法，在每一个数据包的`preprocess()`函数处理之后，通过`combine_data()`函数，对缓存的数据进行统计上的增量数据合并，即完善同一个连接流的统计数据信息和特征。在实例化`ConvNET()`后，将`combine_data()`返回的已经构造好的数据送入`predict()`，运用深度网络进行分类并返回结果。流程如图25所示。



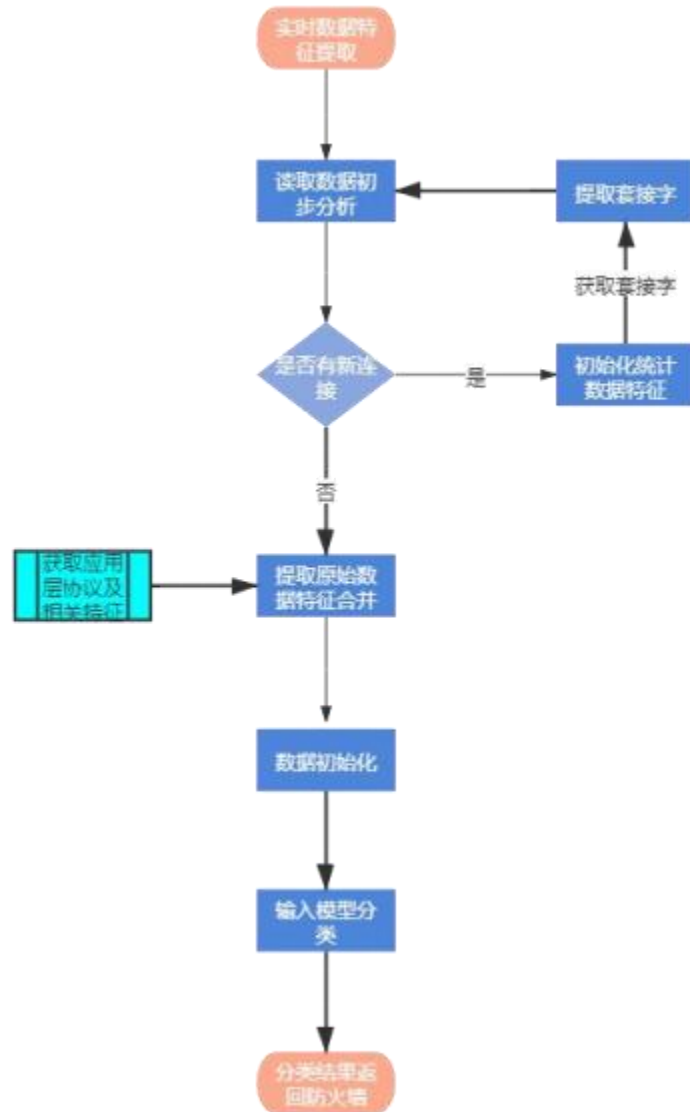


图25 实时数据包的特征提取与分类的主流程

过程中使用 `judge_services()` 获取应用层协议及相关特征，使用 `get_pred_data()` 进行分类数据构造，使用 `data_assign()` 调整数据类型。最后输出分类结果返回给SDN防火墙。如果有新连接进入，则使用 `init_pool()` 初始化统计信息，使用 `get_is_sm_ips_ports()` 获取套接字特征。

### 3.4 管理系统实施方案

后台信息传递。规则信息为csv格式，而前后端信息的交互往往采用json格式。因此，在后台，首先采用 `csv2json` 库将csv转换为json格式，并采用 `HttpResponse` 方法将json格式内容以纯文本的形式传递给前端。前端使用

Bootstrap-Tale插件通过Ajax获取json格式的数据，并将其以表格形式展示给用户。

前端信息传递。在用户在网页端实时编辑信息点击提交后，前端将内容以表单的形式采用POST方法将内容传递给后台。Django后台获取相应的内容后，若是增加内容，则直接添加信息并自动为该信息添加ID索引。若是删除，修改信息，则以ID作为索引对json文件进行查找，并进行相应的操作。

## 4 运行结果/应用效果

系统首页展示系统运行防护时间、新增过滤规则、俘获的数据包、处理的历史DDos攻击威胁等信息。并以图表的信息更加清晰的展示给用户相关历史数据流。

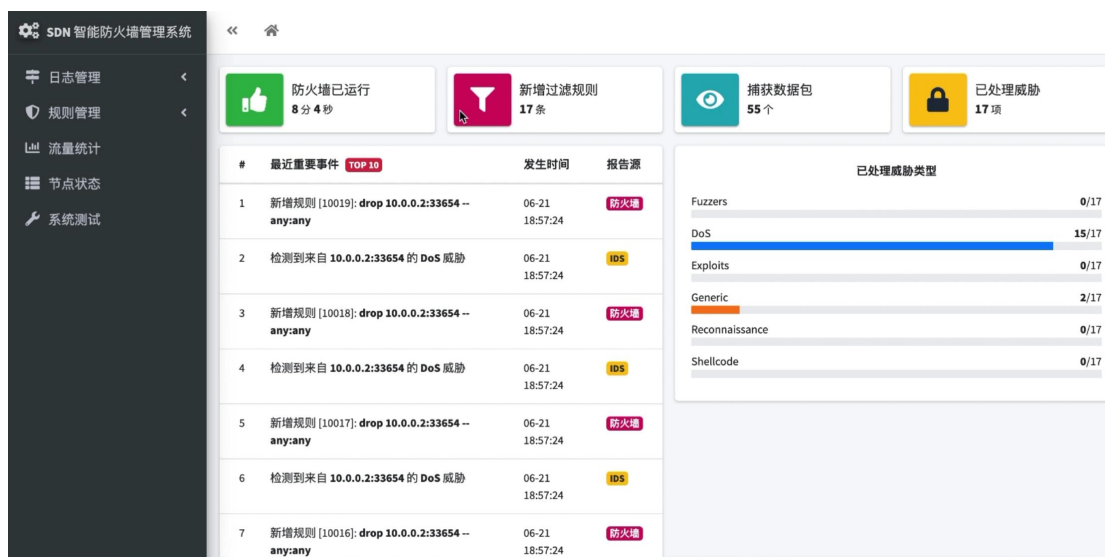
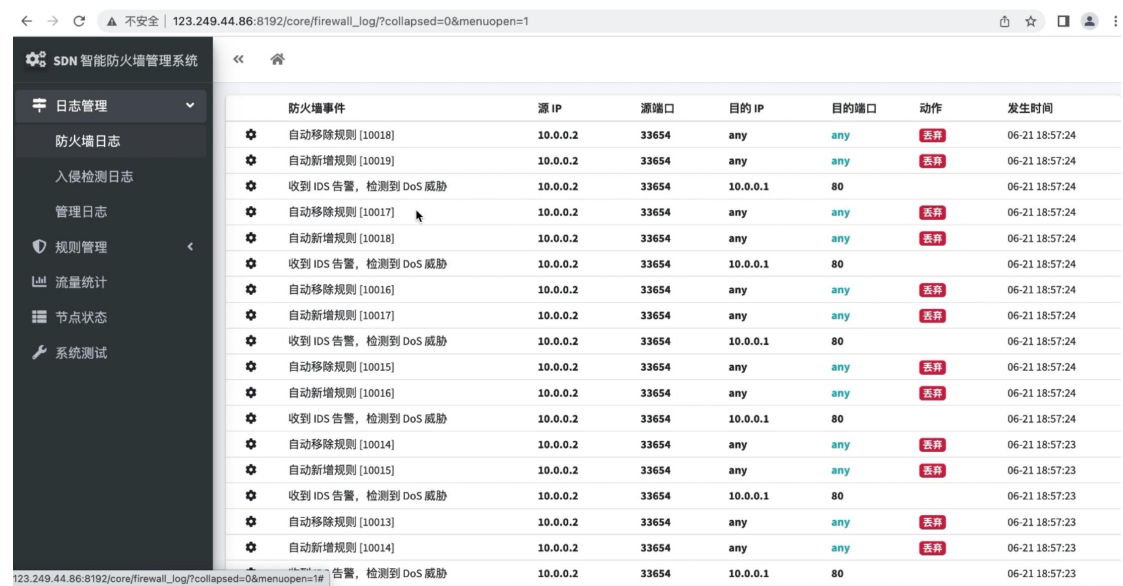


图26 系统首页

下面分别以日志查看、规则编辑、系统测试说明系统的运行和演示效果。

### 4.1 日志查看

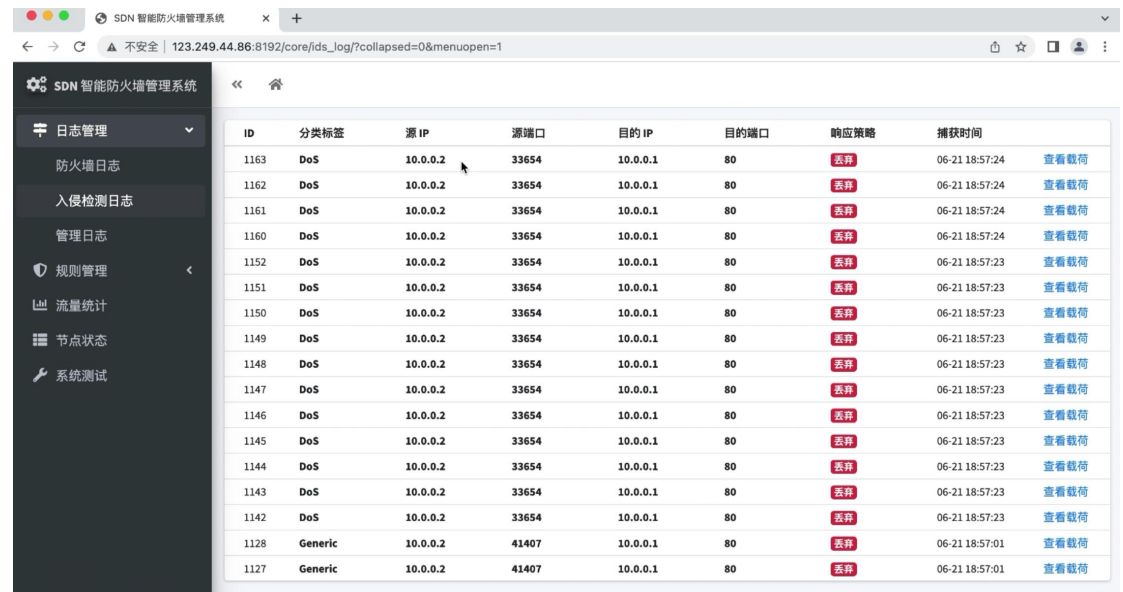
SDN防火墙日志：记录防火墙处理的数据包，内容包括数据包的源IP、目的ip、源端口、目的端口、响应动作、时间等，以时间顺序呈现给用户。



防火墙事件	源 IP	源端口	目的 IP	目的端口	动作	发生时间
自动移除规则 [10018]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
自动新增规则 [10019]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
收到 IDS 告警, 检测到 DoS 威胁	10.0.0.2	33654	10.0.0.1	80		06-21 18:57:24
自动移除规则 [10017]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
自动新增规则 [10018]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
收到 IDS 告警, 检测到 DoS 威胁	10.0.0.2	33654	10.0.0.1	80		06-21 18:57:24
自动移除规则 [10016]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
自动新增规则 [10017]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
收到 IDS 告警, 检测到 DoS 威胁	10.0.0.2	33654	10.0.0.1	80		06-21 18:57:24
自动移除规则 [10015]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
自动新增规则 [10016]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:24
收到 IDS 告警, 检测到 DoS 威胁	10.0.0.2	33654	10.0.0.1	80		06-21 18:57:24
自动移除规则 [10014]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:23
自动新增规则 [10015]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:23
收到 IDS 告警, 检测到 DoS 威胁	10.0.0.2	33654	10.0.0.1	80		06-21 18:57:23
自动移除规则 [10013]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:23
自动新增规则 [10014]	10.0.0.2	33654	any	any	丢弃	06-21 18:57:23
收到 IDS 告警, 检测到 DoS 威胁	10.0.0.2	33654	10.0.0.1	80		06-21 18:57:23

图27 SDN防火墙日志

DDos攻击检测日志：对检测到的攻击进行记录并展示，记录相关攻击的源ip和源端口，以及攻击类型和处理方式和时间。一方面可以防止和避免再次收到类似攻击，另一方面可以为以后作攻击数据处理分析。



ID	分类标签	源 IP	源端口	目的 IP	目的端口	响应策略	捕获时间	
1163	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:24	<a href="#">查看载荷</a>
1162	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:24	<a href="#">查看载荷</a>
1161	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:24	<a href="#">查看载荷</a>
1160	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:24	<a href="#">查看载荷</a>
1152	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1151	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1150	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1149	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1148	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1147	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1146	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1145	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1144	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1143	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1142	DoS	10.0.0.2	33654	10.0.0.1	80	丢弃	06-21 18:57:23	<a href="#">查看载荷</a>
1128	Generic	10.0.0.2	41407	10.0.0.1	80	丢弃	06-21 18:57:01	<a href="#">查看载荷</a>
1127	Generic	10.0.0.2	41407	10.0.0.1	80	丢弃	06-21 18:57:01	<a href="#">查看载荷</a>

图28 DDos攻击检测日志

## 4.2 规则编辑

SDN防火墙规则：展示防火墙的规则界面，并为用户提供新增，修改和删除等功能。

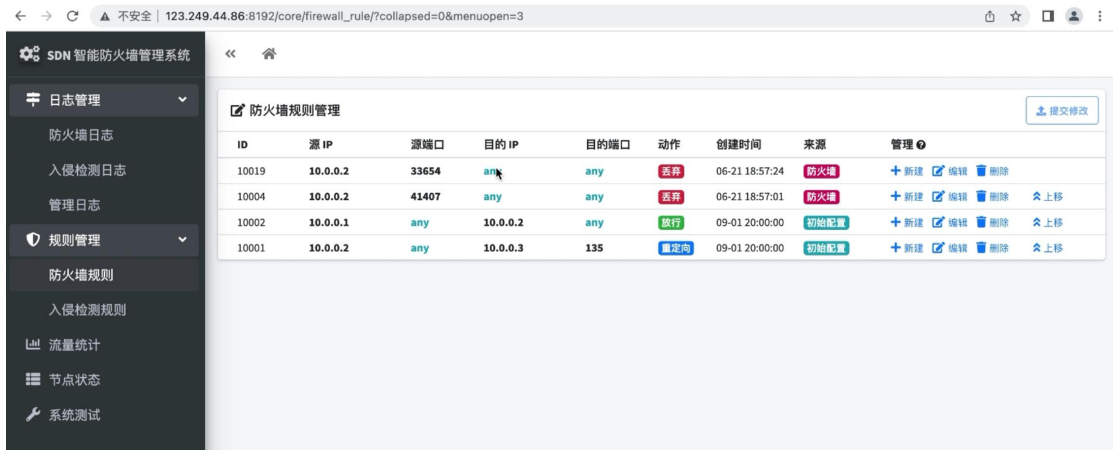


图29 SDN防火墙规则

DDos入侵检测规则：包括7种DDos攻击的分类标签，相关描述，响应策略，历史捕获数量等。提供用户编辑更新功能。

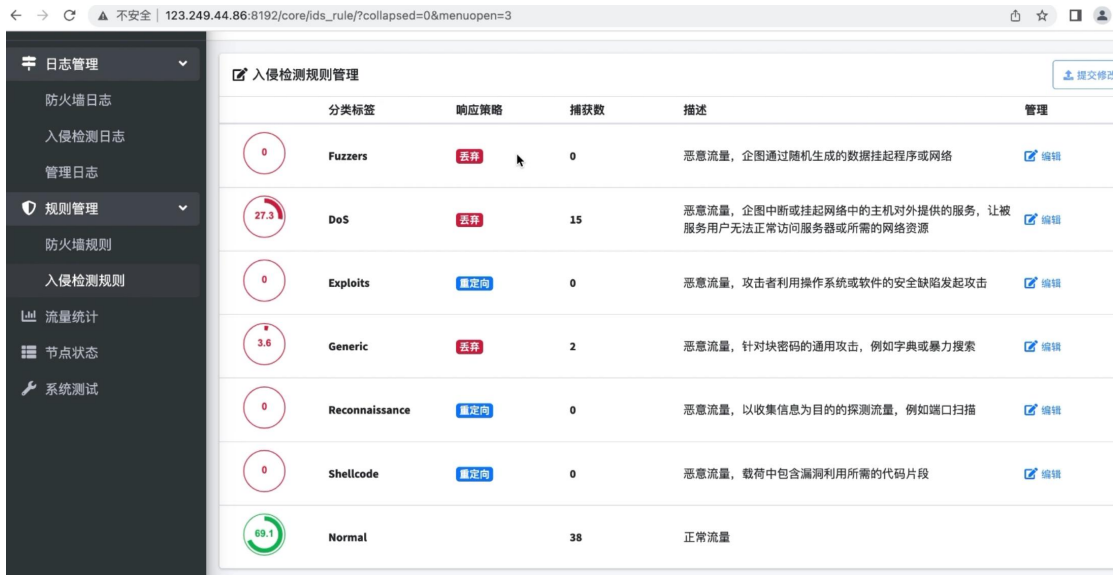


图30 DDoS入侵检测规则

### 4.3 系统测试

链路时延带宽监测系统如下，其中第三列是时延，第四列是带宽，他们是随着系统实时动态变化的

src	dst	delay	bw
6<-->5	:	0.5596	0.0010
1<-->5	:	0.5745	0.0010
1<-->2	:	0.6897	0.0010
5<-->1	:	0.5745	0.0010
5<-->6	:	0.5596	0.0010
2<-->1	:	0.6897	0.0010

src	dst	delay	bw
6<-->5	:	0.3707	0.0010
1<-->5	:	0.4147	0.0010
1<-->2	:	0.5603	0.0010
5<-->1	:	0.4147	0.0010
5<-->6	:	0.3707	0.0010
2<-->1	:	0.5603	0.0010

图31 链路时延带宽监测

发送安全流量包:

```
mininet> h2 ./run.sh safeflow
#### pcap file contains 10 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
mininet>
```

图32 safeflow

可以看到，系统中均判定为正常

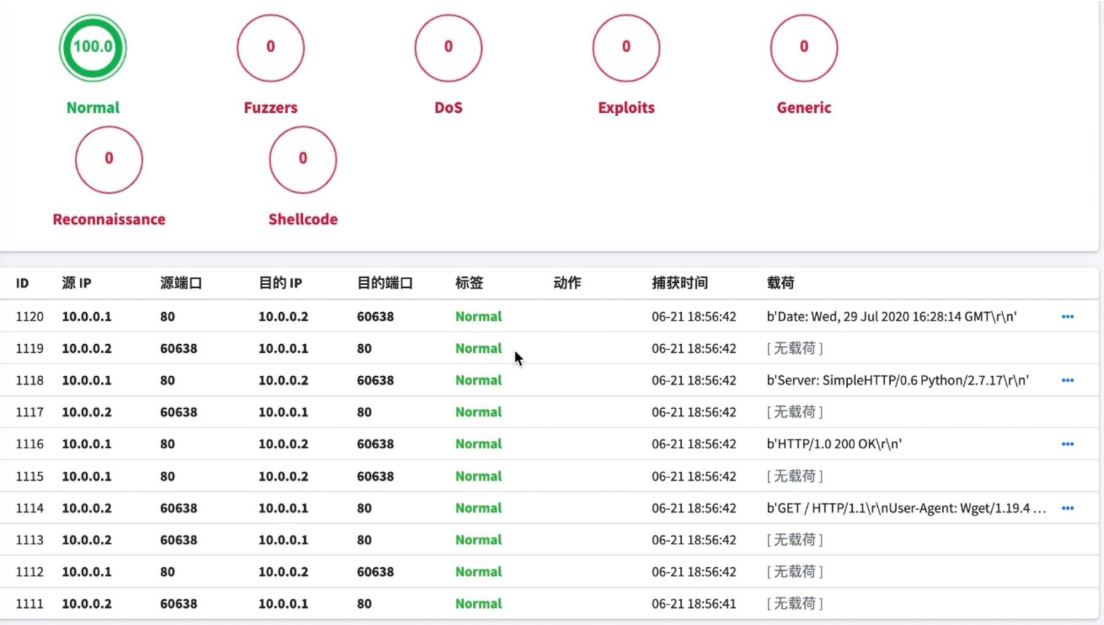


图33 Normal

发送恶意流量:

```
mininet> h2 ./run.sh genattk
#### pcap file contains 20 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
```

图34 genattk

检测成功:

1128	10.0.0.2	41407	10.0.0.1	80	Generic	丢弃	06-21 18:57:01	b'GET /q4qGQ6 HTTP/1.1\r\nHost: PEpvlQQzbu\r...
1127	10.0.0.2	41407	10.0.0.1	80	Generic	丢弃	06-21 18:57:01	b'GET /q4qGQ6 HTTP/1.1\r\nHost: PEpvlQQzbu\r...
1126	10.0.0.2	41407	10.0.0.1	80	Normal		06-21 18:57:01	[ 无载荷 ]

图35 dection



## 5 创新与特色

本作品实现了面向DDos入侵检测的基于深度学习的SDN智能检测防御系统，创新点与特色包括：

**一、基于SDN架构实现防火墙系统。**SDN架构“转控分离”的特点有利于防火墙规则的快速部署和实时更新，“控制集中”的特点有利于对网络内部流量和边界流量实施统一管理。本作品在SDN架构下实现防火墙系统，具备实时、动态更新防火墙规则的能力。并进一步通过SDN防火墙处理DDos入侵检测器检测到的DDos攻击流量。

**二、集成了防火墙、入侵检测、网络时延带宽在线监测系统。**除了传统的结合深度学习的入侵监测系统，我们还集成了监控系统实时监测网络中链路的时延和交换机各个端口的带宽占用情况，在面对大规模网络攻击时，往往需要采取综合防御手段和监测技术才能有效应对。对于网络流量的实时监测，提升了对系统问题的判断预测能力，有助于更好地发现潜在的风险隐患。

**三、在SDN应用中引入基于深度学习的DDos入侵检测技术。**本作品使用以CNN与GRU的混合的深度学习模型训练得到DDos入侵检测器，通过入侵检测器实时监听网络中的流量变化，并及时做出回应。不依赖外部特征库，我们的模型在最新的DDos2019数据集上进行训练，学习数据集的特征，进一步丰富模型，在已有的流量数据集上进行训练，具备检测未知攻击流量的能力。引入基于深度学习的入侵检测技术，有利于发挥SDN动态部署与全局控制的优势，为网络领域的安全控制问题提供动态化、智能化的较为新颖的方案尝试。

**四、提供易用的管理界面。**本作品提供web端的管理界面，可查询告警日志信息，编辑入侵检测规则及防火墙规则，维护效率高、实用性强，并留有一定的扩展空间。本作品拟在此基础上实现更多管理功能，进一步增强系统的易用性和实用性，具体内容可参见第6节。

## 6 后续优化

本作品拟实现一个易用，可靠，具备轻量级、实时性、智能化DDos攻击检测与防御能力的SDN智能检测系统。为此，在目前已有的实现基础上，拟作如下改进：



**在SDN防火墙方面**，拟优化控制台日志，向管理系统提供有关报文和告警数据的更多信息，优化报文及日志存储。拟将部分硬编码参数改进为可配置。拟编写更多自动化测试脚本，进一步测试系统可靠性和稳定性。

**在DDos入侵检测器方面**，进一步改进模型，包括训练数据针对当前网络进行针对性(当前网络环境)初始化、丰富模型结构，尝试在模型中加入残差网络(Resnet)来减小模型因参数过多可能带来的过拟合、梯度爆炸等影响。优化实时特征提取方式，拟实验更多特征组合，进一步拟合源数据集特征，通过额外的软硬件设备监视网络整体情况，提取整体网络环境特征，优化网络结构，进一步提高检测实时性和准确率。

**在管理系统方面**，进一步提升和优化管理系统的各项功能，拟增加更丰富的状态查询功能，如查询网络拓扑、交换机流表项等重要信息。拟提供对系统关键参数的配置界面。拟对管理界面进一步优化，提供更易使用和更具观赏性的交互界面。拟对管理系统进一步测试界面的稳定性，包括点击测试、功能测试、异常排除等。