计算机科学
COMPUTER SCIENCE

SDN中基于统计与集成自编码器的DDoS攻击检测模型

李春江, 尹少平, 池浩田, 杨静, 耿海军

引用本文

李春江, 尹少平, 池浩田, 杨静, 耿海军. SDN中基于统计与集成自编码器的DDoS攻击检测模型[J]. 计算机科学, 2024, 51(11): 389-399.

LI Chunjiang, YIN Shaoping, CHI Haotian, YANG Jing, GENG Haijun. DDoS Attack Detection Model Based on Statistics and Ensemble Autoencoders in SDN [J]. Computer Science, 2024, 51(11): 389-399.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230900028

SDN 中基于统计与集成自编码器的 DDoS 攻击检测模型

李春江[1] 尹少平[1] 池浩田[1] 杨 静[1,3] 耿海军[1,2,3]
1 山西大学自动化与软件学院 太原 030006
2 山西大学计算机与信息技术学院 太原 030006
3 山西大学大数据科学与产业研究院 太原 030006
(chunjiangli18@163.com)

摘 要 软件定义网络(Software-defined Networking,SDN)是一种提供细颗粒集中网络管理服务的新型网络体系结构,主要有控制与转发分离、集中控制和开放接口基本特征。SDN 由于控制层的集中管理逻辑,控制器被攻击者作为理想的分布式拒绝服务攻击(Distributed Denial-of-Service,DDoS)目标。然而,传统的基于统计的 DDoS 攻击检测算法常存在误报率高、阈值固定等问题;基于机器学习模型的检测算法常存在计算资源消耗大、泛化性差等问题。为此,文中提出了一种基于统计特征与集成自编码器的 DDoS 攻击双层检测模型。基于统计的方法提取 Rényi 熵特征,设置动态阈值判断可疑流量;基于集成自编码器算法对可疑流量进行更精确的 DDoS 攻击判断。双层检测模型不仅提升了检测效果,解决了误报率高的问题,同时还有效地缩短了检测时间,从而减少了计算资源的消耗。实验结果表明,该模型在不同网络环境下都有较高的准确率,不同数据集检测的 F1 值最低都达到了 98.5%以上,表现出了很强的泛化性。

关键词:软件定义网络;分布式拒绝服务攻击;Rényi 熵;动态阈值;自编码器
中图分类号 TP393

DDoS Attack Detection Model Based on Statistics and Ensemble Autoencoders in SDN

LI Chunjiang[1], YIN Shaoping[1], CHI Haotian[1], YANG Jing[1,3] and GENG Haijun[1,2,3]
1 School of Automation and Software Engineering, Shanxi University, Taiyuan 030006, China
2 School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China
3 Industry of Big Data Science and Industry, Shanxi University, Taiyuan 030006, China

Abstract Software-defined networking (SDN) is a novel network architecture that provides fine-grained centralized network management services. It is characterized by control and forwarding separation, centralized control, and open interface characteristics. Due to the centralized management logic of the control layer, controllers have becom the prime targets for distributed denial-of-service (DDoS) attacks. Traditional statistics-based DDoS attack detection algorithms often have problems such as high false-positive rates and fixed thresholds, while detection algorithms based on machine learning models are often involved in substantial computational resource consumption and poor generalization. To address these challenges, this study proposes a two-tier DDoS attack detection model based on statistical features and ensemble autoencoders. The statistics-based method extracts Rényi entropy features and sets a dynamic threshold to judge suspicious traffic. The ensemble autoencoder algorithm is then applied for a more accurate DDoS attack judgment of suspicious traffic. The double-layered model not only enhances detection performance and solves the problem of high false alarm rates, but also effectively shortens the detection time, thereby reducing the consumption of computational resources. Experimental results show that the model achieves high accuracy in different network environments, with the lowest F1 score on various datasets is more than 98.5%, demonstrating a strong generalization capability.

Keywords Software-defined networking, Distributed denial-of-service (DDoS), Rényi entropy, Dynamic threshold, Autoencoder

到稿日期:2023-09-04 返修日期:2024-03-03
基金项目:山西省应用基础研究计划(20210302123444);山西省高等学校科技创新项目(2022L002);中国高校产学研创新基金项目(2021FNA02009);国家自然科学基金(61702315);山西省重点研发计划(201903D421003,202202020101004);国家重点研发计划(2018YFB1800401)
This work was supported by the Fundamental Research Program of Shanxi Province (20210302123444), Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (2022L002), Ministry of Education (CN) Industry-University-Research Innovation Fund (2021FNA02009), National Natural Science Foundation of China (61702315), Key Research and Development Program of Shanxi Province (201903D421003, 202202020101004) and National Key Research and Development Program of China (2018YFB1800401).
通信作者:耿海军(genghaijun@sxu.edu.cn)

1 引言

软件定义网络(SDN)是一种新型网络架构[1],它将控制平面与数据平面分离,以软件编程的方式实现对网络的定义和控制。SDN 整合了控制平面,从而使得数据平面中分散在各个网络的设备可以被集中化管理[2]。SDN 架构思想迅速发展并被应用在网络虚拟化[3-4]、5G[5]、数据中心网络[6]、云计算[7]等领域。然而,SDN 的分布式结构存在很多安全漏洞[8],攻击者会针对不同层面进行威胁和攻击[9],其安全性已经成为研究的热点。

拒绝服务攻击(DDoS)具有攻击成本低、攻击效果明显等特点,一直是互联网用户面临的最常见、影响较大的网络安全威胁之一[10-11]。DDoS 攻击的目的是通过发送过多的请求,消耗攻击目标的可用资源,从而导致机器无法提供服务[12]。SDN 环境中,攻击者针对 SDN 架构设计了不同类型的 DDoS 攻击[13]:针对数据层的交换机攻击会造成交换机流表过载、缓冲饱和;针对控制层发送不匹配报文会消耗内存、CPU 资源,造成控制器资源饱和;针对应用层的耗尽北向接口资源攻击或者发起传统应用层 DDoS 攻击会造成网络资源耗尽。由于 SDN 网络控制层具有集中控制管理特点和数据层的转发设备简易,攻击者常常选择控制器作为理想的攻击目标[14]。因此,DDoS 的威胁在 SDN 中比传统网络更加严重,研究 SDN 环境下的 DDoS 攻击检测与防御技术是至关重要的[15]。

SDN 能够实现全局网络流量监测,编写自定义程序控制网络行为。网络管理员可以通过在控制层编写应用程序,对网络流量进行异常检测、动态更新流表规则、快速响应 DDoS 攻击等操作,实现 DDoS 攻击检测与防御。然而,DDoS 攻击具有很高的伪装能力,很难被单一规则检测。近年来,SDN 网络中 DDoS 攻击检测方法主要分为基于统计与基于机器学习的方法[16-17]。基于统计的检测算法是一种轻量级的检测方法,具有实时性,不需要构建较多的流量特征,检测速度快。常用的基于统计的检测方法有基于信息熵[18]、傅里叶分析[19]和小波分析[20]。基于统计的检测算法常采用阈值来判断网络攻击,面对不断变化的网络环境无法准确识别 DDoS 攻击,误报率高[21]。因此,基于统计的检测算法适合设置为初级检测;基于机器学习的检测方法可以从历史数据中自主学习并识别以前未知的攻击,并具有较高的准确性。因此,机器学习算法也越来越多地被用于 DDoS 攻击检测。基于机器学习的 DDoS 攻击检测的算法大致分为 3 类。1)监督学习。监督学习算法可以在标记数据上进行训练,以区分正常和异常的流量模式。决策树、随机森林和支持向量机(SVM)等监督学习算法已被用于检测 SDN 网络中的 DDoS 攻击[22]。2)无监督学习。无监督学习算法可被用于检测 DDoS 攻击,仅使用输入数据来学习数据的结构和特征。这种无需标记数据的学习方式使得无监督学习适用于大规模、复杂的数据集。k-均值聚类和自编码器等无监督学习算法已被用于检测 SDN 网络中的 DDoS 攻击。3)深度学习。深度学习作为机器学习的分支,在处理高维度、非线性、大规模数据时有很好的表现,是研究的热点之一。卷积神经网络(CNN)和递归神经网络(RNN)等深度学习算法已被用于检测 SDN 网络中的 DDoS 攻击。然而,基于机器学习的 DDoS 攻击检测方法也面临多重技术挑战:一方面,基于机器学习的 DDoS 攻击检测系统需要大量的训练数据集,并且容易受到对抗性攻击;另一方面,基于机器学习的 DDoS 攻击检测系统需要占用较大的计算资源,特别是深度学习,由于模型和算法复杂,因此更需要大量的计算资源,显著增加了计算成本。总体而言,基于机器学习的 DDoS 攻击检测技术在 SDN 环境下具有广泛的应用前景且面临多重技术挑战。

本文结合了统计方法和机器学习方法,设计了 SDN 环境下 DDoS 攻击双层检测模型。基于统计的方法使用了更能体现正常流量和 DDoS 流量差异的 Rényi 熵特征作为网络流量初步检测对象[22],设置动态的阈值来满足 100% 召回率和低误报率。基于机器学习的方法使用复杂度低的自编码器算法,自编码器算法适用于在线异常检测的无监督神经网络,检测效率高。自编码器由两部分组成:编码器和解码器。编码器学习特征向量映射到隐藏层,解码器重构特征数据,最后根据均方根误差(Root Mean Square Error,RMSE)计算重构误差。集成自编码器是将多个自编码器分为集成层和输出层。集成层由多个自编码器有序组成,每个自编码器独立计算出 RMSE 值;输出层的自编码器可当作集成的非线性投票机制,学习集成层输出的 RMSE,最终将生成的 RMSE 值作为异常判断依据。使用集成自编码器的原因是:1)它可以以无监督的方式进行训练,学习少量网络流量也能有很好的检测效果,适用于实时监测;2)集成自动编码器可以通过组合多个自动编码器的输出来提高异常检测的准确性,提高模型的鲁棒性,降低过拟合的风险[23];3)自编码器分为训练部分和执行部分,可以将训练部分和基于统计检测方法的动态阈值学习部分结合,加强模型的自动化学习能力。

综上所述,本文在 SDN 环境中提出了一个 DDoS 攻击双层检测模型,主要贡献包括 4 个方面。

1)设计了双层检测模型。使用 Rényi 熵统计特征作为第一层检测模块的标准,动态阈值的设定有效降低了误报率,缓解了基于机器学习方法的计算资源消耗大的问题。

2)模型自适应能力增强。模型将 Rényi 熵动态阈值的学习更新部分和自编码器训练部分结合,使模型能自适应于不同的网络环境。

3)提取了针对 DDoS 攻击和 SDN 网络的特征向量。通过手动提取 15 维的特征向量作为集成自编码器检测模型的输入,与传统的机器学习输入特征相比,有效地提高了检测的准确率。

4)设计了轻量级 DDoS 攻击检测模型。本文使用了集成自编码器对数据进行无监督学习,提高了模型的检测和泛化能力。轻量级模型可适用于在线实时检测,对于少量的训练集也有不错的检测效果。

2 相关工作

随着 SDN 网络的发展,DDoS 攻击利用网络设备和网络架构发起攻击,对 SDN 环境的可用性和安全性造成了严重威胁[24]。近年来,SDN 网络中的 DDoS 攻击检测是国内外的热门研究领域,研究人员和网络服务供应商在 SDN 网络环境

中提出了多种基于统计、机器学习和混合方法的 DDoS 攻击检测技术。

Tsobdjou 等[25]提出了一种基于熵的在线 DDoS 攻击检测系统，在客户端与服务端通信过程中实时检测 DDoS 攻击。其关键在于利用归一化熵进行异常攻击检测，并根据正常流量的增长更新阈值。Ahalawat 等[26]提出了一种基于 Rényi 熵 DDoS 的检测方法。该方法采集了流量的统计信息，并创建哈希表用于分析和计算 Rényi 熵值；通过截取一定窗口大小的流量计算 Rényi 熵，并根据预先设定的阈值判断其是否为 DDoS 攻击，若连续 5 次超出阈值，则判断为异常，进行 DDoS 攻击缓解。然而，基于统计的方法依赖于预设的阈值来识别 DDoS 攻击，这可能会导致误报或漏报。如果阈值设置过高，则可能会漏报真正的 DDoS 攻击；反之，如果设置过低，则可能会引发误报，认为正常流量是攻击。尽管 Tsobdjou 等的方法根据正常流量的增长更新阈值，但这种更新机制可能无法及时适应突发性和动态的网络流量变化。当 Rényi 熵连续 5 次超出阈值时，Ahalawat 等的方法才判断受到 DDoS 攻击，这可能会导致检测的延迟。

Fouladi 等[27]提出了一种 SDN 环境下基于时间序列分析的 DDoS 攻击检测模型。该方案采用流量特征预测、混沌理论模型、指数滤波器和动态阈值方法来检测网络中的实时变化。时间序列分析可以更好地捕捉到网络流量的动态变化，提供更准确的预警，但时间序列模型的参数选择和调整要求较高。Isa 等[28]提出了一种混合深度自编码器和随机森林分类器模型，以增强 SDN 环境中的入侵检测性能。混合的模型可以结合两种算法的优点，提高检测的准确性，但需要大量的数据和计算资源。基于机器学习的检测算法准确率高、误报率低，但存在检测速率低、CPU 占用率高等问题。尤其是深度学习，当网络流量过大时会影响检测时间，增加计算成本。

Tan 等[29]在数据平面部署了 DDoS 攻击检测的触发机制来筛查网络中的异常流量。该机制利用流量的速率特征和不对称特征初步判断，然后使用基于 k-Means 和 k-邻近（k-Nearest Neighbors，KNN）的组合机器学习算法检测可疑流量。该方法能够保障大规模网络流量下的 DDoS 检测效率和网络质量。Wang 等[30]提出了一种基于信息熵和深度学习的 DDoS 攻击检测方法。首先，控制器可以通过信息熵来检测可疑流量。然后，由 CNN 模型执行基于数据包的细粒度检测，以区分正常流量和攻击流量。类似地，Zhang 等[31]提出的 DDoS 攻击检测模型通过计算网络流量的信息熵，快速找出可能存在异常的流量，然后利用深度神经网络（Deep Neural Network，DNN）模型对这些可疑流量进行深入分析，以判断其是否真的是 DDoS 攻击。两种方法使用信息熵快速筛选出可能的异常流量，对异常流量进行更精确的检测，提高了检测速率。这些方法大多需要大量有标签的数据集进行训练，模型训练时间较长；实验检测部分选用的数据集单一，多为本地模拟的流量，对算法的泛化能力没有评估。

通过以上的研究发现，基于统计的检测算法检测速率快、准确率较高，但无法作为可靠的检测模型，多用于 DDoS 攻击异常触发机制。本文针对上述研究方法中出现的问题，提出了一种基于统计与机器学习的双层检测方法，对流量进行特征提取，使用 Rényi 熵设置合适的动态阈值区间，触发基于集成自编码器的检测模型，提升模型的检测速度和准确率。

3　基于统计与集成自编码器的 DDoS 攻击检测方法

3.1　攻击检测模型框架

本文基于机器学习算法设计了集成自编码器，并结合统计检测方法，提出了基于统计与集成自编码器的 DDoS 攻击检测模型，如讨图 1 所示。DDoS 攻击检测模型的核心功能部署在 SDN 网络的控制器端，该模型主要包括两个模块：基于统计特征异常检测模块和基于集成自编码器检测模块。

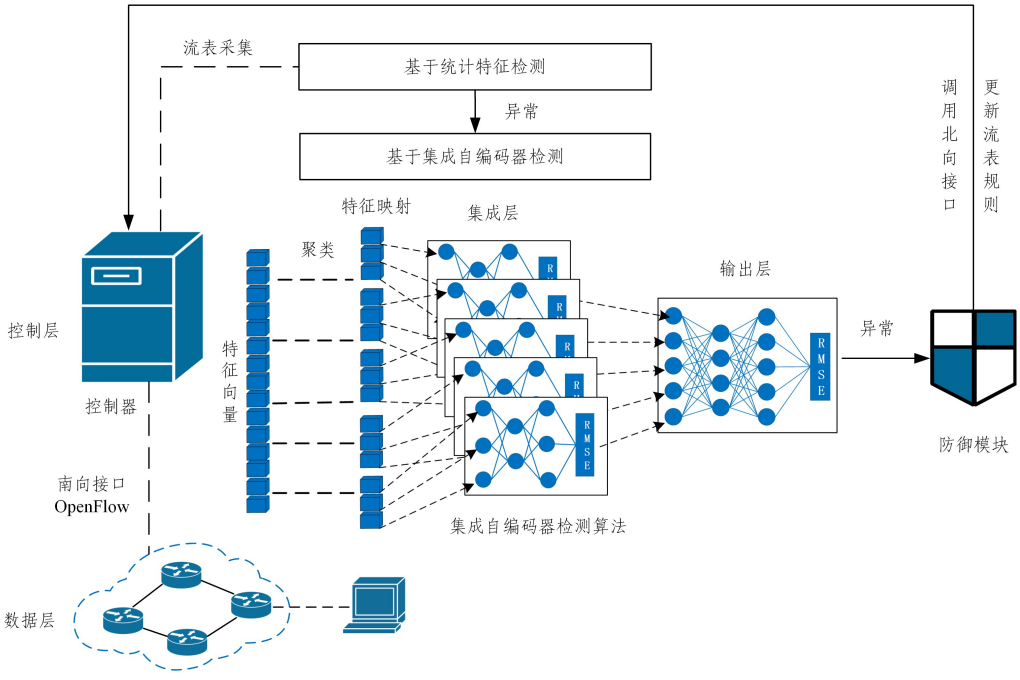

图 1　基于统计与集成自编码器的 DDoS 攻击检测模型的架构图

Fig. 1　Architecture diagram of DDoS attack detection model based on statistics and ensemble autoencoder

为了更好地解释模型检测原理，本文基于 OpenFlow 通信协议下的工作流程详细说明 SDN 中基于统计与集成自编码器的 DDoS 攻击检测过程。

检测模型在执行过程中通过滑动窗口的方式利用控制器读取数据层交换机中的流表，获取流表中的匹配字段和计数器；模型将采集的数据流信息交付给基于统计特征检测模块进行源 IP(Source IP Address，SrcIP)和目的 IP(Destination IP Address，DstIP)特征提取，计算 SrcIP 和 DstIP 的 Rényi 熵。若 Rényi 熵超过阈值范围，则初步认为网络异常，将数据流传递给第二层检测模块；基于集成自编码器检测模块接收到异常信号，从数据流中细颗粒地提取 15 维特征向量。特征向量根据相关性聚类分为 5 个子向量，映射到集成自编码器中进行更准确的检测。如果检测结果为异常，防御模块会更新流表规则，防御 DDoS 攻击。

本文使用的部分符号定义，如表 1 所列。

表 1 符号定义

Table 1 Symbol definitions

符号	含义
$H_\alpha(X)$	阶数为 α 时变量 X 的 Rényi 熵
μ_t	t 时刻的数据平均值
σ_t^2	t 时刻的数据方差
w	滑动窗口大小
S	单次采集数据流总数
$RMSE$	自编码器均方根误差
UCL	动态阈值上线
LCL	动态阈值下线
$\vec{x}$	特征向量
v	有序特征向量集
v_i	第 i 个自编码器特征向量
AE_i	集成层第 i 个自编码器
T_{rmse}	输出层 $RMSE$ 阈值

3.2 基于统计特征检测模型

传统基于统计的方法大多只选择 Shannon 熵作为特征，然而在面对低速率 DDoS 攻击、网络瞬时拥塞等状况时，香农熵的检测算法显得单一且误报率较高。本文选取了 Rényi 熵特征作为网络流量初步检测对象，Rényi 熵相比香农熵更能明显体现不同分布之间的差异，减少初步检测的误报率。

1)统计特征 Rényi 熵

信息熵是对随机变量集合有序化程度的一种度量。集合中随机变量越是有序，信息熵越低；反之，随机变量越混乱，信息熵就越高。Rényi 熵是一种特殊的信息熵，通用地表达了各种熵的概念。对于给定离散随机变量 X 可能出现的结果 $\{x_1, x_2, \cdots, x_n\}$ 和对应的概率 $p_i \in p_1, p_2, \cdots, p_n$，且当阶数 $\alpha \geqslant 0$ 且 $\alpha \neq 1$ 时，Rényi 熵的定义如式(1)所示：

$$H_\alpha(X) = \frac{1}{1-\alpha} lb\left(\sum_{i=1}^{n} p_i^\alpha\right) \tag{1}$$

随着阶数的不同，Rényi 熵形式也不相同。

当 $\alpha \to 1$ 时，$H_\alpha(X)$ 收敛于香农熵：

$$H_1(X) = \lim_{\alpha \to 1} H_\alpha(X) = -\sum_{i=1}^{n} p_i lb\ p_i \tag{2}$$

当 $\alpha = 0$ 时，$H_\alpha(X)$ 得到最大值：

$$H_0(X) = lb(n) \tag{3}$$

当 $\alpha = 2$ 时，Rényi 熵也称碰撞熵，对于混乱的分布，其相比香农熵更能体现熵值的差异，有利于阈值的设定。

$$H_2(X) = -lb\sum_{i=1}^{n} p_i^2 \tag{4}$$

本文选取了 SrcIP 和 DstIP 数据进行 Rényi 熵计算，最终初步异常检测提取的特征如表 2 所列。

表 2 基于统计的特征

Table 2 Statistic based characteristics

特征名	含义
$H(SrcIP)$	源地址的 Rényi 熵
$H(DstIP)$	目的地址的 Rényi 熵

2)阈值设定

传统基于统计方法的检测通常使用固定的阈值，固定的阈值随着网络变化有较大的误差，影响检测效果。本文为了使基于统计特征检测具有自适应性，利用了指数加权移动平均值(Exponentially Weighted Moving-Average，EWMA)思想设置了动态阈值。阈值的设定是以 μ_t 为中心线，取控制上线(Upper Control Limit，UCL)和控制下线(Lower Control Limit，LCL)作为阈值(动态阈值)，计算式如下：

$$UCL = \mu_t + 3\sigma_{\text{EWMA}} \tag{5}$$

$$LCL = \mu_t - 3\sigma_{\text{EWMA}} \tag{6}$$

其中，σ_{EWMA} 是根据 t 时刻之前的数据方差 σ_t^2 计算出的标准差，EWMA 统计量的方差如式(7)所示：

$$\sigma_{\text{EWMA}}^2 = \left(\frac{\lambda}{2-\lambda}\right)\sigma_t^2 \tag{7}$$

动态阈值学习过程中，使用正常数据进行训练，但在执行过程时不更新阈值范围。训练过程中，模型通过不断学习新观测的特征值，实现对阈值范围的动态调整。通过划分训练过程和执行过程，可以使动态阈值与自编码器的训练保持一致，使模型在不同网络环境下的阈值设置更加灵活。其中阈值训练过程如算法 1 所示。

算法 1 阈值训练

输入：数据训练迭代次数 max_iter；数据训练当前迭代次数 cur_iter；当前获取的特征值 H_t；式(5)中的加权因子 $\lambda(0<\lambda\leqslant 1)$

输出：阈值范围的 UCL 和 LCL

```
while cur_iter ≤max_iter do
根据当前获取的H_t更新μ_t和σ_t^2
σ_EWMA^2=(λ(2−λ)) * σ_t^2;/ * 根据新得到的方差，更新σ_EWMA^2 * /
    UCL=μ_t+3 * sqrt(σ_EWMA^2) / * 更新动态阈值的上线 * /
LCL=μ_t−3 * sqrt(σ_EWMA^2) ;/ * 更新动态阈值的下线 * /
cur_iter=cur_iter+1 ;/ * 迭代数加一 * /
    endwhile
return UCL,LCL;/ * 返回阈值区间 * /
```

3.3 特征提取

第一层检测模块检测出 Rényi 熵超过动态阈值区间后，会将数据流传递给第二层检测模块进行更准确的检测。机器学习算法中，特征的选取直接影响着算法的性能，针对检测目标提取的特征能够增强检测的准确率，适量的特征数目能够降低算法的复杂度。本文受到 Wang 等[32]提出的六元组特征向量的启发，设计了更详细的特征向量作为输入特征。

在特征提取中，使用滑动窗口方式提取流表信息中匹配字段和计数器内的通信标识数据。当采集的流数目达到窗口大小 w 时，进行特征提取。滑动窗口方法的优点是在处理实时数据时，随着新数据的到来，旧数据将从窗口中移出，从而减少了内存占用和计算资源。提取所需的特征向量后，

输入到集成自编码器检测模型，用于训练和执行。集成自编码器检测模型的输入特征如表 3 所列。

表 3 集成自编码器的输入特征

Table 3 Input features of ensemble autoencoders

特征名	含义
Interval	时间间隔
$H(SrcIP)$	源地址的 Rényi 熵
$H(DstIP)$	目的地址的 Rényi 熵
$H(SrcPort)$	源端口的 Rényi 熵
$H(DstPort)$	目的端口的 Rényi 熵
PRF	可逆流百分比
$GRIF$	不可逆流增长率
$PFSP$	小数据包流百分比
$PFSD$	短持续时间流百分比
GRF	流的增长速率
AFP	流的平均数据包
$AFBS$	流的平均每秒字节数
AFPS	流的平均每秒数据包数
AFD	流的平均持续时间
Protocols	协议总数

时间间隔 Interval：样本采集第一个流和最后一个流的时间差，在 DDoS 攻击时，时间间隔非常小。$H(SrcPort)$ 和 $H(DstPort)$ 熵值的计算方式与 SrcIP 和 DstIP 相同，如式(4)所示。

可逆流百分比(Percentage of Reversible Flows，PRF)：在交换机流表项中任意两个流满足 SrcIP 和 DstIP 互逆，且通信协议相同，则定义这两个流为可逆流(Reversible Flows，RF)。大多数 DDoS 攻击伪造 SrcIP，发送大量无用的数据包，使得服务器无法回溯建立双向连接。因此，在 DDoS 攻击过程中，采集的数据流总数目 S 不变，采集的可逆流(RFsum)减少，导致 DDoS 攻击时可逆向流百分比显著下降。PRF 的计算式如下：

$$PRF=\frac{RFsum}{S} \tag{8}$$

其中，采集数据流的总数为滑动窗口大小，即 $S=w$。

不可逆流增长率(Growth Rate of Irreversible Flows，GRIF)：其在遭受 DDoS 攻击时网络流量特征与可逆流百分比情况相反，会急剧增加。GRIF 的计算式如下：

$$GRIF=\frac{S-RFsum}{Interval} \tag{9}$$

小数据包流百分比(Percentage of Flows with a Small Number of Packets，PFSP)：DDoS 攻击时通常会在短时间内产生大量的流，每个流中所包含的数据包非常少。因此，当小数据包流总数 FSP_{sum} 在采集数据流总数中占比增加时，PFSP 的增大。PFSP 的计算式如下：

$$PFSP=\frac{FSP_{\text{sum}}}{S} \tag{10}$$

短持续时间流百分比(Percentage of Flows with a Short Time Duration，PFSD)：DDoS 大部分报文都是无效的，控制器响应不会持续很长时间。因此，当短持续时间流总数 FSD_{sum} 在采集数据流总数中占比增加时，PFSD 增大。PFSD 的计算式如下：

$$PFSD=\frac{FSD_{\text{sum}}}{S} \tag{11}$$

流增长速率(Growth Rate of Flows，GRF)：其表示数据流数目增长速度，其计算式如下：

$$GRF=\frac{S}{Interval} \tag{12}$$

数据包数、字节数是网络通信流量的重要特征之一。在 DDoS 攻击中，流的平均数据包(Average Packets of Flows，APF)、流的平均每秒字节数(Average Bytes per Second of Flows，ABSF)、流的平均每秒数据包(Average Packets per Second of Flows，APSF)和流的平均持续时间(Average Duration of Flows，ADF)会和正常网络有差异。当 $item\in\{packets, packetspersecond, bytespersecond, duration\}$ 时，求平均值的通用计算式为：

$$Average=\frac{\sum_{0}^{S} item}{S} \tag{13}$$

综上所述，本文根据流表信息结合 DDoS 攻击特点设计了维度为 15 的特征向量 $\vec{x}$，将其作为基于集成自编码器检测模型的输入特征，来提高检测模型的性能。

3.4 集成自编码器检测模型

自编码器(Autoencoder，AE)是一种无监督的神经网络模型，主要用于数据的降维、特征提取和数据生成等任务。自编码器通常由两部分组成：编码器(Encoder)和解码器(Decoder)。编码器将原始数据编码为一个较低维度的隐藏表示，而解码器则将这个隐藏表示解码为一个尽可能接近原始数据的重构。定义输入特征向量为 $\vec{x}$，编码器通过激活函数 f、权重矩阵 $\boldsymbol{W}_{\text{enc}}$ 和偏置向量 $\vec{b}_{\text{enc}}$ 得到编码特征 $\vec{h}$，计算过程为：

$$\vec{h}=f(W_{\text{enc}}\vec{x}+\vec{b}_{\text{enc}}) \tag{14}$$

解码器输出需要将 $\vec{h}$ 传递到解码器，解码器使用激活函数 g、权重矩阵 $\boldsymbol{W}_{\text{dec}}$ 和偏置向量 $\vec{b}_{\text{dec}}$ 计算得到重构样本 $\vec{y}$，计算过程为：

$$\vec{y}=g(W_{\text{dec}}\vec{h}+\vec{b}_{dec}) \tag{15}$$

自编码器的核心思想就是学习输入特征向量之间的关系，实现重构的数据与原始数据分布恒等，达到 $\vec{y}\approx\vec{x}$。自编码器在异常检测任务中分为训练阶段和执行阶段，在训练阶段学习正常的网络流量，并在执行阶段检测与正常网络流量不符的异常流量。其原理是正常网络流量训练得到的自编码器能够较好地重构与训练数据分布相同的数据，而对于异常分布的数据，重构误差会较大。因此，通过比较输入实例与其重构之间的 RMSE，可以判断实例是否为异常。当 $\vec{x}$ 的维度为 n 时，$\vec{y}$ 和 $\vec{x}$ 之间的 RMSE 的损失函数如下：

$$RMSE(\vec{x},\vec{y})=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-y_i)^2} \tag{16}$$

在训练过程中，使用随机梯度下降算法(Stochastic Gradient Descent，SGD)进行参数优化，通过反向传播计算损失误差调整参数 $W\in W_{enc}$，W_{dec} 和 $\vec{b}\in\vec{b}_{enc}$，$\vec{b}_{dec}$，更新计算式如下：

$$W=W-l\frac{\partial \text{RMSE}}{\partial W} \tag{17}$$

$$\vec{b}=\vec{b}-l\frac{\partial \text{RMSE}}{\partial \vec{b}} \tag{18}$$

其中，l 是学习率，可以控制参数更新幅度。

特征向量 $\vec{x}$ 在进入神经网络之前会将 15 个特征根据相关性距离进行层次聚类，并转换为树形结构。然后，这个树形结构被递归地分解，以确保每个分支聚类的特征数尽量达到 5。至此，15 个特征向量被分为 5 组，映射到 5 个更小的子实例

特征中，构成一个有序集 v。

$$v=\{\vec{v_1},\vec{v_2},\cdots,\vec{v_5}\} \tag{19}$$

集成自编码器算法由集成层 L_1 和输出层 L_2 组成。

集成层：5 个自编码器组成的有序集合为集成层，每个自编码器处理 v 中的相应实例。在训练模式下，集成层中自编码器学习它们各自子空间的正常行为，计算损失函数关于权重的梯度，并根据梯度信息更新自编码器参数。在训练模式和执行模式下，每个自动编码器都将其 RMSE 报告给输出层。

输出层：输出层由一个自编码器组成，它接收 k 个自编码器的 RMSE 值作为输入。在训练阶段，输出层的自编码器学习集成层自编码器的 RMSE 之间的关系，以减少网络流量中的噪声影响。在执行阶段，输出层生成准确可靠的异常分数，以实现异常检测的目的。

集成自编码器算法如图 2 所示，模型的训练过程和执行过程如算法 2 和算法 3 所示。

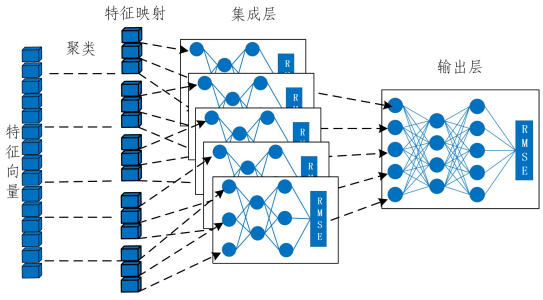

图 2 集成自编码器算法的结构

Fig. 2 Structure of ensemble autoencoder algorithm

算法 2 集成自编码器训练算法

输入：数据训练迭代次数 max_iter；数据训练当前迭代次数 cur_iter；特征向量集 v；第 i 个自编码器的参数 AE_i；训练集最大特征值和最小特征值 max_v，min_v

输出：训练后的 L_1，L_2

1. / * 训练输入层 L_1 * /
2. while cur_iter＜max_iter do
3. z=zeros(5)；/ * 定义一个长度为 5 的零向量，存储 L_1 的 RMSEs * /
4. for AE_i in[1,5] do / * 对集成层的每一个自编码器进行训练 * /
5. 记录输入特征 $\vec{v}$ 的最大值和最小值，更新 max_v 和 min_v
6. $\vec{v_i}=norm_{0-1}(\vec{v_i})$；/ * 根据 max_v 和 min_v，归一化处理 * /
7. 输入 $\vec{v_i}$，利用式(14)和式(15)得到重构向量 $\vec{y_i}$；
8. z[i]←RMSE(v_i', y_i)；/ * 计算重构误差，保存到 z * /
9. 反向传播，利用式(17)和式(18)更新计算权值 W_i 和偏置 $\vec{b_i}$
10. end for
11. / * 训练输出层 L_2 * /
12. $\vec{z'}=norm_{0-1}(\vec{z})$；/ * 归一化集成层输入的 RMSE 向量 * /
13. 输入 $\vec{z'}$，利用式(14)和式(15)得到重构向量 $\vec{y}$；
14. 反向传播，利用式(17)和式(18)更新计算权值 W 和偏置 $\vec{b}$
15. cur_iter=cur_iter+1
16. endwhile
17. return L_1，L_2

算法 3 集成自编码器执行算法

输入：H(SrcIP)，H(DstIP)，LCL 和 UCL；特征向量集 v；第 i 个自编码器的参数 AE_i

输出：输出层的 RMSE

1. / * 集成层 L_1 执行算法 * /
2. if H(SrcIP) or H(DstIP) not in [LCL, UCL] do/ * 当 Rényi 熵不在动态阈值范围内，执行 * /
3. z=zeros(5)；
4. for AE_i in[1,5] do
5. $\vec{v_i}=norm_{0-1}(\vec{v_i})$；
6. 输入 $\vec{v_i}$，利用式(14)和式(15)得到重构向量 $\vec{y_i}$；
7. z[i]←RMSE(v_i', y_i)；
8. end for
9. / * 输出层 L_1 执行算法 * /
10. $\vec{z'}=norm_{0-1}(\vec{z})$；
11. 输入 $\vec{z'}$，利用式(14)和式(15)得到重构向量 $\vec{y}$；
12. RMSE=RMSE($\vec{z'}$, $\vec{y}$)；
13. endif；
14. if H(SrcIP) and H(DstIP) in [LCL, UCL] do / * 当 Rényi 熵都满足于动态阈值范围，执行 * /
15. RMSE=0.0；/ * 不执行集成自编码器算法，RMSE 设为 0 * /
16. endif；
17. return RMSE；

输出层在执行算法中输出的 $RMSE\in[0,+\infty)$，被作为 DDoS 攻击检测的判断标准，RMSE 越接近 0 表示数据和正常数据分布越一致。本文提出的检测模型是基于自编码器算法，DDoS 攻击根据网络监测中实际输出的 RMSE 值来进行判断。然而，模型的性能需要评估指标来进行衡量，设置合适的阈值对于模型检测性能的衡量至关重要。自编码器常用的方法有基于最大误差：将训练阶段对正常流量数据的最大 RMSE 值乘以一个超参数 $\beta(\beta>1)$ 作为阈值，β 越大，检测标准越严格。本文采用基于统计分析的方法：计算正常数据的 RMSE 值的均值 μ 和标准差 σ，通过均值加上标准差设计阈值 T_{rmse}，具体计算式如下：

$$T_{rmse}=\mu+p\sigma \tag{20}$$

其中，p 是一个经验参数，可以根据实际应用场景和误报率要求进行调整。

4 公开数据集上的实验结果及分析

4.1 实验环境和数据集

实验基于 Linux 环境下的 Pytorch 深度学习框架，Python 版本为 3.8，Cuda 版本为 11.4，采用 NVIDIA Tesla V100S-PCIE-32GB 进行加速。为了验证模型的泛化性和可靠性，实验选用了 CIC-IDS2017，CSE-CIC-IDS2018，CIC-DDoS2019 和 InSDN 数据集。

CIC-IDS2017 数据集由加拿大网络安全研究中心(Canadian Institute for Cybersecurity，CIC)发布，包含了大量正常网络流量和常见攻击的记录。数据集模拟了真实世界的网络环境，使用 CICFlowMeter 对网络流量进行分析。流量记录包括时间戳、IP、端口、协议和持续时间等信息，以便研究人员对网络入侵行为进行识别和分析。CIC-IDS2017 数据集中包含多种攻击类型，其中包括 DoS 和 DDoS 攻击。CSE-CIC-IDS2018 和 CIC-IDS2017 类似，但它包含数百万个网络流量记录，涵盖正常网络流量和恶意攻击流量，可为研究人员提供

丰富的数据资源。CIC-DDoS2019 数据集同样由 CIC 发布，主要关注 DDoS 攻击，该数据集包含多种 DDoS 攻击类型的记录，如 UDP Flood，HTTP Flood，SYN Flood 等，但是官网未提供正常数据集。InSDN 数据集是专门针对 SDN 网络环境而设计的数据集，包括了 SDN 在数据平面的 DDoS 和 DoS 攻击以及控制平面的 DDoS 和 DoS 攻击。

数据集采集的初始特征信息如表 4 所列，各数据集提取的流数目如表 5 所列。其中，CSE-CIC-IDS2018 正常数据集和 CIC-DDoS2019 的 DDoS 攻击数据集只截取了部分数据。

表 4　数据集初始特征

Table 4　Initial features of dataset

特征名	特征描述
Src Port	源端口号
Dst IP	目的 IP 地址
Dst Port	目的端口号
Protocol	协议类型
Timestamp	时间戳
Flow Duration	流持续时间
Tot Fwd Pkts	正向数据包总数
Tot Bwd Pkts	反向数据包总数
Flow Byts/s	流平均字节每秒传输速率
Flow Pkts/s	流平均包每秒传输速率

表 5　各数据集中的流数目

Table 5　Number of flows in each dataset

数据集	正常流数目	DoSS 攻击流数目
CIC-IDS2017	2 271 300	379 738
CSE-CIC-IDS2018	4 000 000	549 840
CIC-DDoS2019	0	4 000 000
InSDN	68 425	127 120

4.2　实验评估指标

实验使用召回率 Recall 和误报率（False Alarm Rate，FAR）作为第一层基于统计检测模块的评价指标。

Recall 指实际为 DDoS 流量且被模型预测为 DDoS 流量的比例，定义为：

$$Recall=\frac{TP}{TP+FN} \tag{21}$$

误报率（FAR）指将正常流量误报为 DDoS 攻击流量的比例，定义为：

$$FAR=\frac{FP}{FP+TN} \tag{22}$$

其中，TP 表示真正例数（即实际为 DDoS 流量且被模型正确预测为 DDoS 流量的数目）；FN 表示假反例数（即实际为 DDoS 流量但被模型错误预测为正常流量的数目）；FP 表示假正例数（即实际为正常流量但被模型错误预测为 DDoS 流量的数目）；TN 表示真反例数（即实际为正常流量且被模型正确预测为正常流量的数目）。

本文使用准确率 Accuracy、召回率 Recall 和 F1 值来评估基于集成自编码器检测模型的性能。

准确率（Accuracy）指模型对所有样本进行分类的正确率，即所有分类正确的样本数除以总样本数。其计算式为：

$$Accuracy=\frac{TP+TN}{TP+FP+TN+FN} \tag{23}$$

Precision 指模型预测为 DDoS 流量且实际为 DDoS 流量的比例，其定义如下：

$$Precision=\frac{TP}{TP+FP} \tag{24}$$

F1 值是 Precision 和 Recall 的调和平均数，是一种综合指标，可以同时考虑精确率和召回率。F1 值越高，模型的性能越好。其通常定义为：

$$F1=2\times\frac{Precision\times Recall}{Precision+Recall} \tag{25}$$

4.3　实验参数

参数调整实验选取了具有大量正常数据的 CSE-CIC-IDS2018 数据集，数据集前 4 000 000 条是正常数据集。实验按照滑动窗口进行特征提取，运行基本参数默认值如表 6 所列。实验主要对 Rényi 熵阶数 α、动态阈值加权因子 λ、滑动窗口大小 w 和训练数据量参数进行调优。

表 6　实验参数设置

Table 6　Experimental parameter settings

参数	参数描述	实验值
α	Rényi 的阶数	2
λ	动态阈值加权因子	0.4
p	T_{rmse} 经验参数	8
w	滑动窗口大小	100
max_iter	训练迭代次数	15 000

1）Rényi 熵阶数 α：由式（2）可知，传统 Shannon 熵即为 $\alpha\to1$ 的 Rényi 熵。实验主要对比 $\alpha\to1$ 和 $\alpha=2$ 的 Rényi 熵，结果如图 3 所示。

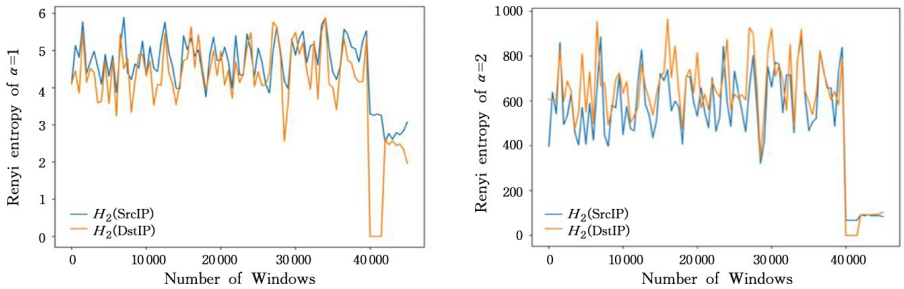

图 3　Rényi 熵阶数对比

Fig. 3　Rényi entropy order comparison

由图 3 可以看出，当 $\alpha=2$ 时，Rényi 熵的范围更大，更能显著区分正常流量和 DDoS 攻击流量。

2）动态阈值加权因子 λ：由式（5）—式（7）可知，λ 越接近 1，动态阈值范围越大。理论上，较小的 λ 值会导致误报率较高，较大的 λ 值可以降低误报率，但对 DDoS 攻击识别的能力也会降低。在数据集 CSE-CIC-IDS2018 上将 λ 增大，动态阈值区间的变化以及基于 Rényi 熵初步检测的 FAR 和 Recall 的结果如图 4 所示。

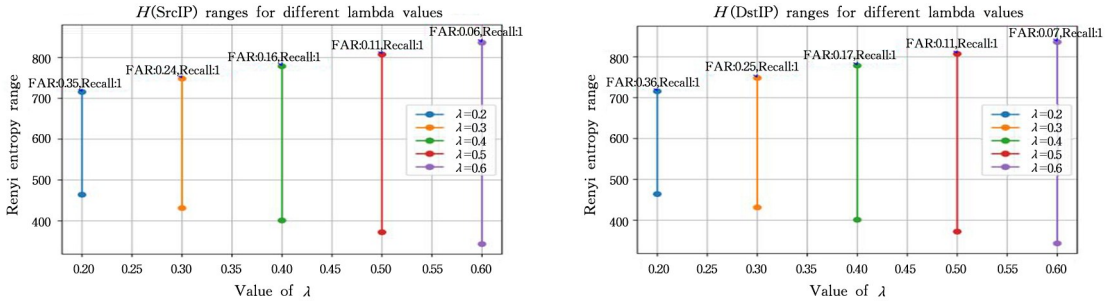

图 4 λ 不同取值结果

Fig. 4 Results with different values of λ

由图 4 可以看出，动态阈值的区间逐渐增大，误报率降低，召回率还能一直保持 100%。这是因为单一的网络环境流量分布比较稳定，Rényi 熵特征值差距大。然而，实际网络环境更加复杂。阈值区间越大，DDoS 攻击漏检的可能性也越大，作为第一层的检测模块需要追求 100%的召回率。因此，实验选择了误报率较低且区间较窄的 $\lambda=0.4$。

检测模型的评估需要对输出的 RMSE 值设置阈值，将其作为二分类的标准，并对 DDoS 攻击流量进行标记。由于不同的网络环境以及不同的学习特征都会影响 RMSE 结果，因此固定的阈值没有评估的意义。本文使用了式(20)计算 CSE-CIC-IDS2018 正常数据集不同滑动窗口下的 T_{rmse}，经过多次实验，确定了参数 $p=8$ 时能最大程度满足不同网络情况的阈值设定。其中，剔除了 RMSE 为 0 的无效值和大于 1 的异常值。

3)不同滑动窗口 w：实验使用不同的滑动窗口 w 的检测结果如图 5 所示，不同滑动窗口下的评估指标效果如表 7 所列。其中，每个图中的虚线表示当前滑动窗口下的 T_{rmse}。

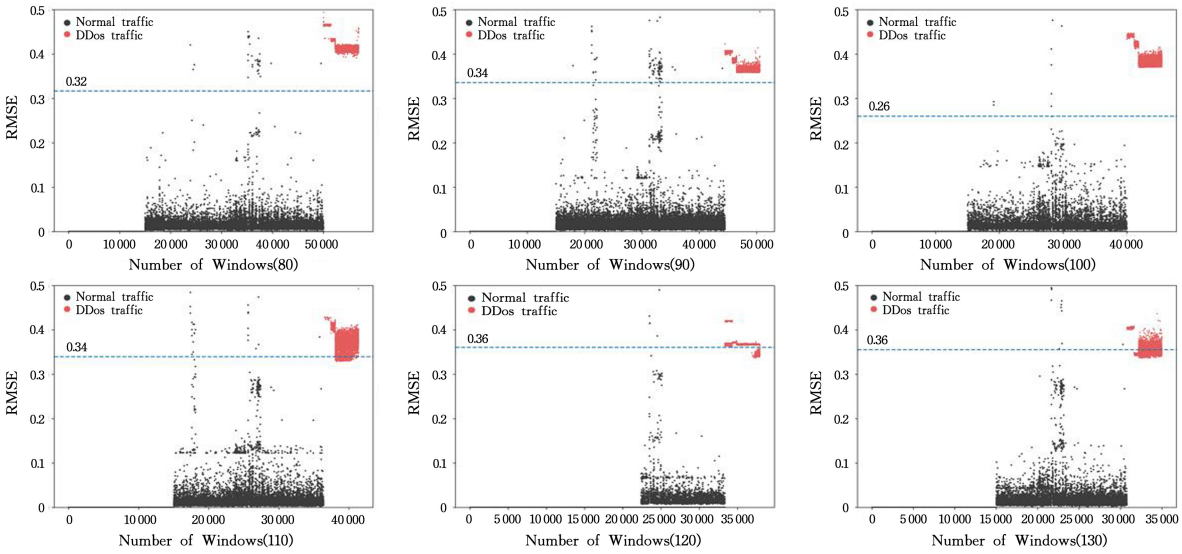

图 5 w 的不同取值结果

Fig. 5 Results with different values of w

由图 5 中可以看出，当 $p=8$ 时，$w=100$ 的 *RMSE* 阈值最小，正常数据更收敛。

表 7 不同 w 下的评估指标

Table 7 Evaluation metrics with different values of w

(%)

窗口大小	*Precision*	*Accuracy*	*Recall*	*F1*
80	99.60	99.65	100	99.80
90	99.37	99.38	100	99.69
100	99.71	99.68	100	99.86
110	99.56	98.86	94.68	97.06
120	99.73	99.21	96.35	97.96
130	99.35	93.71	51.97	68.25

从表 7 可以看出，$w=100$ 时的准确率和 F1 值是最优的，模型综合效果最好。

4)训练数据量：在无监督学习环境中，集成自编码器的表现往往与训练数据量的大小成正比。当训练数据量增加时，由自编码器生成的检测模型的性能通常会得到提升。实验将训练数据量设为 5000～20000，结果如图 6 所示。

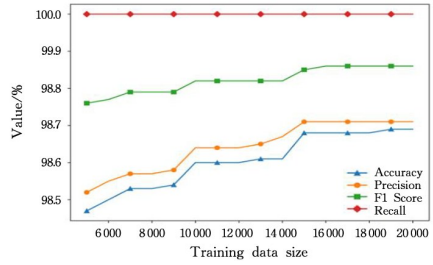

图 6 不同训练数据规模的检测结果

Fig. 6 Detection results with different training data size

由图 6 可以看出，训练数据达到 15000 时，检测性能趋于最优。

4.4 对比实验

1）单双层对比实验

为了展示双层检测模型的效果，设置了单独使用集成自编码器算法的检测模型作为对比实验。CSE-CIC-IDS2018 数据集一共有 4 549 840 条数据流，重复实验了 5 次，运行时间（Time）取平均值，不同检测模型的准确率和 F1 值如表 8 所列。

表 8 单双层对比实验

Table 8 Comparison experiment between single-layer and double-layer detection models

层数	Time/s	*Accuracy*/%	*F1*/%
1	197.12	99.55	99.80
2	142.10	99.68	99.86

由表 8 可以得出，加了基于统计的检测模块后，模型评估的指标得到了提高。其中，运行时间缩短了 55 s，说明该模块对于检测模型的效率有显著的提升作用。

2）数据特征对比实验

为了验证基于滑动窗口手动提取的特征对模型性能的提升，设计了特征如表 3 和表 4 所列的两组实验，实验准确率和 F1 值如表 9 所列。

表 9 数据特征对比实验

Table 9 Data feature comparison experiment

数据特征	Time/s	*Accuracy*/%	*F1*/%
初始特征	396.35	97.43	98.66
手动提取特征	142.10	99.68	99.86

由表 9 可以得出，基于滑动窗口手动提取的特征性能优于传统初始特征。

3）数据集对比实验

根据 3.3 节设置的实验参数，对 3.1 节提出的数据集进行实验。其中，对于正常数据达不到训练数据量 15 000 的数据集，会使用 CSE-CIC-IDS2018 的部分正常数据集进行补充，主要目的是检测不同数据集的 DDoS 攻击流量，实验结果如表 10 所列。

表 10 不同数据集下的评估指标

Table 10 Evaluation metrics on different datasets

（%）

数据集	*Precision*	*Accurac*	*Recall*	*F1*
CIC-IDS2017	97.56	95.37	100	98.76
CSE-CIC-IDS2018	99.71	99.68	100	99.86
CIC-DDoS2019	99.71	99.82	100	99.86
InSDN	99.96	99.96	100	99.98

由表 10 可知，模型在不同的数据集下，对于 DDoS 攻击检测都有较高的准确率。其中 CIC-IDS2017 数据集中包含 DoS slowloris，DoS Slowhttptest，DoS Hulk，DoS GoldenEye 和 DDoS 等多种攻击类型，模型能够达到 100% 的召回率，但在准确度上还有待提升。

4）不同机器学习模型的对比实验

实验选择了文献[31]的基于信息熵和 DNN 的 DDoS 攻击检测模型、传统的 KNN 模型和单个自编码器 AE 模型进行对比，在数据集 CSE-CIC-IDS2018 上的检测结果如表 11 所列。

表 11 不同模型下的评估指标

Table 11 Evaluation metrics on different models

（%）

检测模型	*Precision*	*Accuracy*	*Recall*	*F1*
AE	99.48	97.85	86.35	92.39
KNN	98.55	99.71	99.08	98.82
Shannon 熵＋DNN	97.51	97.44	97.44	97.31
Rényi 熵＋Ensemble of AEs（本文算法）	99.71	99.68	100	99.86

由表 11 可以看到，单个 AE 模型的准确度和召回率明显较低；传统的 KNN 模型总体效果略逊于本文方法；基于 Shannon 熵和 DNN 的检测模型的检测效果不如本文算法。

5 模拟数据集上的实验结果及分析

5.1 实验环境和数据集

为了验证模型在实际网络中的检测能力，使用 Mininet 构建了一个 SDN 网络。控制器为基于 Python 开源的 Ryu 控制器，实验环境是 8 核 CPU 的 Ubantu20.04 操作系统。该网络中，通信协议为 Openflow1.3 协议，交换机为虚拟交换机（open vSwitch，OvS），拓扑图如图 7 所示。

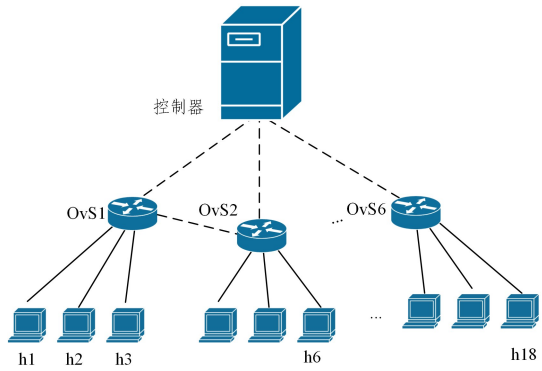

图 7 实验网络拓扑结构

Fig. 7 Experimental network topology

实验使用了 hping3 工具向主机 host1，host3，host4 和 host17 模拟注入了 DDoS 攻击，以从 Ryu 控制器中捕捉交换机流表信息。实验收集了 266 万条数据，其中正常数据有 90 万条，DDoS 攻击流量有 176 万条。

5.2 实验结果

由于数据集限制，实验设置训练数据集为 5 000，其余参数与 3.3 节设置相同，实验结果如图 8 所示。

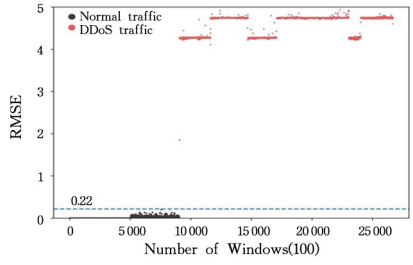

图 8 数据 RMSE 散点图

Fig. 8 RMSE scatter plots

由图 8 可知，正常流量和 DDoS 攻击流量的 RMSE 值有

明显区别。由式(21)求得$T_{rmse}=0.22$,并且 DDoS 攻击检测准确率达到了 100%,说明本文提出的基于统计与集成自编码器的 DDoS 攻击检测模型在 SDN 网络中能够精准地检测 DDoS 攻击流量。

结束语 本文提出了一个基于统计和集成自编码器的 DDoS 攻击检测模型,使用了 Rényi 熵统计特征作为第一层检测模块标准,缓解了基于统计检测误报率高的问题。第二层检测模块中,集成自编码器算法提高了模型的检测能力,适用于在线实时检测,对于少量的训练集也有不错的检测效果。模型将动态阈值的学习更新部分和自编码器训练部分结合,增强了模型的泛化能力,适用于不同的网络环境。

然而,该方法也存在一些不足之处。本文需要对滑动窗口、动态阈值区间和输出层 RMSE 阈值的参数进行调优,而调优过程依赖于有标签的数据集。在今后的工作中,将致力于实现双层无监督的攻击检测模型;在实际应用中,面对复杂变化的网络,能够用轻量级无监督的检测模型实现对 DDoS 攻击的检测和防御。

参 考 文 献

[1] KREUTZ D, RAMOS F M V, VERISSIMO P E, et al. Software-defined networking: A comprehensive survey[J]. Proceedings of the IEEE, 2014, 103(1): 14-76.

[2] FEAMSTER N, REXFORD J, ZEGURA E. The road to SDN: an intellectual history of programmable networks[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(2): 87-98.

[3] ORDONEZ-LUCENA J, AMEIGEIRAS P, LOPEZ D, et al. Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges[J]. IEEE Communications Magazine, 2017, 55(5): 80-87.

[4] YANG M, LI Y, JIN D, et al. OpenRAN: a software-defined ran architecture via virtualization[J]. ACM SIGCOMM computer communication review, 2013, 43(4): 549-550.

[5] TRIVISONNO R, GUERZONI R, VAISHNAVI I, et al. SDN-based 5G mobile networks: architecture, functions, procedures and backward compatibility[J]. Transactions on Emerging Telecommunications Technologies, 2015, 26(1): 82-92.

[6] LI D, CHEN G H, REN F Y, et al. Data Center Network Research Progress and Trends[J]. Chinese Journal of Computers, 2014, 37(2): 259-274.

[7] SON J, BUYYA R. A taxonomy of software-defined networking (SDN)-enabled cloud computing[J]. ACMComputing Surveys (CSUR), 2018, 51(3): 1-36.

[8] WANG M M, LIU J W, CHEN J, et al. Software Defined Networking: Security Model, Threats and Mechanism[J]. Journal of Software, 2016, 27(4): 970-987.

[9] DEB R, ROY S. A comprehensive survey of vulnerability and information security in SDN[J]. Computer Networks, 2022, 206: 108802.

[10] BAWANY N Z, SHAMSI J A, SALAH K. DDoS attack detection and mitigation using SDN: methods, practices, and solutions [J]. Arabian Journal for Science and Engineering, 2017, 42: 425-441.

[11] LIU Z, JIN H, HU Y C, et al. Practical proactive DDoS-attack mitigation via endpoint-driven in-network traffic control[J]. IEEE/ACM Transactions on Networking, 2018, 26(4): 1948-1961.

[12] BHATIA S, BEHAL S, AHMED I. Distributed denial of service attacks and defense mechanisms: current landscape and future directions[J]. Versatile Cybersecurity, 2018: 55-97.

[13] KAUR S, KUMAR K, AGGARWAL N, et al. A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions[J]. Computers & Security, 2021, 110: 102423.

[14] REVATHI M, RAMALINGAM V V, AMUTHA B. A machine learning based detection and mitigation of the DDOS attack by using SDN controller framework[J]. Wireless Personal Communications, 2022, 127(3): 2417-2441.

[15] TAYFOUR O E, MARSONO M N. Collaborative detection and mitigation of DDoS in software-defined networks[J]. The Journal of Supercomputing, 2021, 77: 13166-13190.

[16] NOORIBAKHSH M, MOLLAMOTALEBI M. A review on statistical approaches for anomaly detection in DDoS attacks[J]. Information Security Journal: A Global Perspective, 2020, 29(3): 118-133.

[17] JIA K, WANG J N, LIU F. DDoS detection and mitigation Framework in SDN[J]. Journal of Cyber Security, 2021, 6(1): 17-31

[18] WANG R, JIA Z, JU L. An entropy-based distributed DDoS detection mechanism in software-defined networking[C]// 2015 IEEE Trustcom/BigDataSE/ISPA. IEEE, 2015, 1: 310-317.

[19] LIU Z, HU C, SHAN C. Riemannian manifold on stream data: Fourier transform and entropy-based DDoS attacks detection method[J]. Computers & Security, 2021, 109: 102392.

[20] FOULADI R F, ERMIŞ O, ANARIM E. A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN[J]. Computer Networks, 2022, 214: 109140.

[21] ZHAO P, ZHAO W T, FU Z J, et al. SDN self-protection system based on Renyi entropy[J]. Chinese Journal of Network and Information Security, 2021, 7(3): 85-94.

[22] SWAMI R, DAVE M, RANGA V. Defending DDoS against software defined networks using entropy[C]// 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). IEEE, 2019: 1-5.

[23] MIRSKY Y, DOITSHMAN T, ELOVICI Y, et al. Kitsune: an ensemble of autoencoders for online network intrusion detection [J]. arXiv: 1802.09089, 2018.

[24] CHETOUANE A, KAROUI K. A survey of machine learning methods for DDoS threats detection against SDN[C]// Distributed Computing for Emerging Smart Networks: Third International Workshop, DiCES-N 2022, Bizerte, Tunisia. Springer International Publishing, 2022: 99-127.

[25] TSOBDJOU L D, PIERRE S, QUINTERO A. An online entropy-based DDoS flooding attack detection system with dynamic threshold[J]. IEEE Transactions on Network and Service Management, 2022, 19(2): 1679-1689.

[26] AHALAWAT A, BABU K S, TURUK A K, et al. A low-rate

DDoS detection and mitigation for SDN using Rényi Entropy with Packet Drop[J]. Journal of Information Security and Applications，2022，68：103212.

[27] FOULADI R F，ERMIŞ O，ANARIM E. A DDoS attack detection and defense scheme using time-series analysis for SDN[J]. Journal of Information Security and Applications，2020，54：102587.

[28] ISA M M，MHAMDI L. Hybrid Deep Autoencoder with Random Forest in Native SDN Intrusion Detection Environment[C]// ICC 2022－IEEE International Conference on Communications. IEEE，2022：1698-1703.

[29] TAN L，PAN Y，WU J，et al. A new framework for DDoS attack detection and defense in SDN environment[J]. IEEE Access，2020，8：161908-161919.

[30] WANG L，LIU Y. A DDoS attack detection method based on information entropy and deep learning in SDN[C]// 2020 IEEE 4th Information Technology，Networking，Electronic and Automation Control Conference(ITNEC). IEEE，2020，1：1084-1088.

[31] ZHANG L，WANG J S. DDoS Attack Detection Model Based on Information Entropy and DNN in SDN[J]. Journal of Computer Research and Development，2019，56(5)：909-918.

[32] WANG T，GUO Z，CHEN H，et al. BWManager：Mitigating denial of service attacks in software-defined networks through bandwidth prediction[J]. IEEE Transactions on Network and Service Management，2018，15(4)：1235-1248.

LI Chunjiang，born in 1998，master candidate. His main research interests include anomaly traffic detection and software defined networking.

GENG Haijun，born in 1983，Ph.D，associate professor. His main research interests include network architecture and routing algorithm.

（责任编辑：何杨）

CCF 西南办事处办公区正式启用

10 月 19 日，CCF 西南办事处举办办公区启用仪式。CCF 成都、重庆、绵阳、长沙、昆明、南宁、贵阳等会员活动中心代表，艺术分会、CTO CLUB 上海、宁波运营中心等会员代表与秘书长唐卫清一起见证了办事处办公区的启用。

CCF 西南办事处是 CCF 成立的第二个地方办事处，选址落户于位于成都市青羊区的数字金融大厦。未来 CCF 西南办事处将以服务会员为首位宗旨，持续提升会员服务质量，为 CCF 会员在西南打造一个家，一座桥，做好学会在西南地区的计算机科技工作者的服务工作，加强与地方产业和学术界的合作与交流，促进政产学研的融合发展，为推动区域经济的发展做出更大的贡献。

同期，CCF 西南区域发展研讨会在办事处举行，绵阳主席张晖、贵阳主席彭长根、长沙老主席满君丰、重庆秘书长武春岭、南宁秘书长胡小春、昆明监委主席潘文林、成都副主席章乐、成都副主席徐震等西南各会员活动中心及 YOCSEF 分论坛 AC 代表参与会议。大家针对如何更好地扩大 CCF 在西南区域的影响力展开了热烈的讨论，拟定于 2025 年 7 月中下旬在成都举办 CCF 政产学研用发展大会。与会代表纷纷表示，西南办事处办公区的启用，让西南各会员活动中心在成都有了个“家”，不仅加强了彼此之间的联系，也有利于促进业务上的深入合作。

据 CCF 微信公众号