基于深度学习的SDN异常流量分布式检测方法

王坤[1,2]，付钰[1]，段雪源[3,4]，俞艺涵[5]，刘涛涛[1]

（1.海军工程大学信息安全系，湖北 武汉 430033；2.信阳职业技术学院信息与通信工程学院，河南 信阳 464000；
3.信阳师范大学计算机与信息技术学院，河南 信阳 464000；4.河南省教育大数据分析与应用重点实验室，河南 信阳 464000；
5.海军工程大学作战运筹与规划系，湖北 武汉 430033）

摘　要：针对传统异常流量检测方法在执行大规模软件定义网络（SDN）的检测任务时，存在运算开销大、共享链路繁忙，容易引起网络设备单点故障，导致软件定义网络服务质量下降甚至网络瘫痪等问题，提出一种基于深度学习的SDN异常流量分布式检测方法。该方法将部署在云端服务器的判别器与若干部署在SDN控制器的生成器构造为“一对多”的分布式生成对抗网络（D-VAE-WGAN），利用正常流量样本完成对D-VAE-WGAN的协同训练，在控制器上生成具有独立检测功能的异常流量检测代理，以实现大规模SDN环境下各控制器子网中异常流量的分布式检测。实验结果表明，该方法可以快速、准确地检测出大规模SDN中的异常样本，在准确率、召回率等检测指标上优于传统方法；并且具备对未知异常的检测能力。

关键词：深度学习；软件定义网络；分布式；异常流量检测

中图分类号：TN915.08

文献标志码：A

DOI：10.11959/j.issn.1000−436x.2024199

Distributed abnormal traffic detection method for SDN based on deep learning

WANG Kun[1,2], FU Yu[1], DUAN Xueyuan[3,4], YU Yihan[5], LIU Taotao[1]

1. Department of Information Security, Naval University of Engineering, Wuhan 430033, China
2. School of Information and Communication Engineering, Xinyang Vocational and Technical College, Xinyang 464000, China
3. College of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China
4. Henan Key Laboratory of Analysis and Applications of Education Big Data, Xinyang 464000, China
5. Department of Operational Research and Programming, Naval University of Engineering, Wuhan 430033, China

Abstract: Addressing the high computational expenses, congested shared links, and propensity for single-point failures in network devices that can lead to a degradation of software defined network (SDN) service quality or even network paralysis during the execution of large-scale SDN detection tasks by traditional abnormal traffic detection methods, a distributed abnormal traffic detection method for SDN based on deep learning was proposed. This method constructed a “one-to-many” distributed generative adversarial network (D-VAE-WGAN) with a discriminator deployed on a cloud server and multiple generators deployed on SDN controllers. Utilizing normal traffic samples, collaborative training of the D-VAE-WGAN was completed, resulting in independent abnormal traffic detection proxies on controllers, enabling distributed detection of abnormal traffic within each controller's subnet in a large-scale SDN environment. Experimental results indicate that this method can rapidly and accurately detect abnormal samples in large-scale SDN, outperforming traditional methods in detection metrics such as accuracy and recall rate, and can detect unknown anomalies.

Keywords: deep learning, software defined network, distributed, abnormal traffic detection

收稿日期：2024–09–03；修回日期：2024–11–05

通信作者：俞艺涵，chenyike1992@163.com

基金项目：国家自然科学基金资助项目（No.62102422）；河南省科技攻关基金资助项目（No.242102211070）

Foundation Items：The National Natural Science Foundation of China (No.62102422), Henan Province Key Science and Technology Research Projects of China (No.242102211070)

0　引言

互联网、云计算、大数据技术的深度融合，需要更加安全、稳定、可靠的高性能通信基础设施作为支撑。然而，随着网络规模的不断扩大和用户数量的持续增多，传统的组网模式和管理方法已无法满足互联网发展的现实需求。软件定义网络（SDN，software defined network）是一种将网络设备的控制与转发功能解耦，使网络独立于硬件设备发展的新型网络架构[1]，主要包括控制平面、数据平面和应用平面，如图1所示。

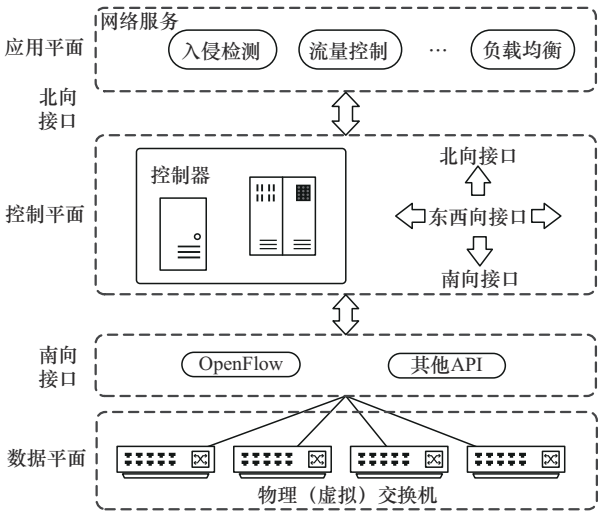


图1　SDN架构

控制平面是SDN的核心，包括控制器及网络操作系统，主要负责制定转发任务、调试通信单元，执行应用平面下发的指令，对网络资源进行优化配置等。数据平面也称转发平面，由交换机、路由器等网络转发设备组成，主要通过南向接口接收控制平面制定的转发规则，并根据规则处置网络流。应用平面主要是各种面向业务的网络应用，如入侵检测、流量控制、负载均衡等应用，它们通过北向接口把需要执行的网络控制行为下发给控制器。

与传统网络架构相比，SDN使用基于虚拟化网络资源设备的操作系统，将网络流量控制和网络设备管理等功能集中在控制器上，在数据平面只保留简单的数据包转发功能[2]。这使得SDN可以根据实际应用需求，对网络布局进行灵活调整。然而，SDN的集中式控制和开放性接入也给网络带来诸多安全隐患。以当前流行的OpenFlow协议交换机为例，当交换机接收到任何新的网络数据流（在交换机流表中无匹配规则的流）时，都会向控制器发送消息询问转发规则，这给攻击者发起针对控制器或交换机的拒绝服务（DoS, denial of service）攻击[3]、分布式拒绝服务（DDoS, distributed denial of service）攻击[4]以及低速率拒绝服务（LDoS, low rate denial of service）攻击[5]带来极大便利，此外还容易引发针对SDN系统漏洞的蠕虫病毒[6]等网络攻击。这些攻击行为不仅会对SND的正常运行造成不良影响，甚至会威胁到核心控制器，导致整个SDN瘫痪。

研究人员希望通过对SDN中网络流量的检测和分析，来发现网络中的异常行为，为进一步采取应对措施，增强网络的稳健性提供支撑。然而将传统的基于统计分析[7]、基于信息论[8]、基于聚类分析[9]和基于机器学习[10]的集中式检测方法迁移到多控制器的大规模SDN中，势必会对通信链路的关键节点和执行集中检测任务的网络设备带来巨大压力，从而极易引发单点故障，甚至造成SDN无法正常运行[11]。本文中的大规模SDN是指由云端服务器通过共享链路连接的多个控制器及所属子网组成的网络模式。因此，融合了深度学习的分布式检测方法，能够在多个网络节点上分配检测任务，实现检测负载的均衡化配置，逐渐成为大规模SDN安全防护领域研究热点。

深度学习具有强大表征能力，不受特征工程的约束，在机器视觉、自然语言处理、金融数据分析预测等复杂任务中有着广泛应用。本文将变分自编码器（VAE, variational auto encoder）[12]与生成对抗网络（GAN, generative adversarial network）[13]相结合提出基于分布式VAE与GAN（D-VAE-WGAN, Distributed VAE-WGAN）架构的大规模SDN中网络异常流量的分布式检测方法，该方法将模型训练与异常检测控制在不同的网络范围内进行。模型训练阶段由云端服务器和各控制器等关键的网络节点参与运算，节点间通过共享链路进行数据交换；异常检测阶段仅由各控制器上的检测代理独立地完成所属子网流量数据的分析与异常检测，控制器与云端服务器间、各控制器之间无检测数据流转。这种设计可以减轻检测阶段云端服务器的计算压力和共享链路的带宽消耗，并且子网间互不干扰；配合控制器的调度功能可将异常流量限制在子

网内部，从而防止网络威胁扩散。本文的贡献可概括如下。

1) 提出一种基于深度学习的SDN异常流量分布式检测方法，将异常检测任务分配给每个SDN控制器所属的子网，避免集中式检测方式引发单点故障的风险。

2) 设计出一种“一对多”的分布式生成对抗网络架构模型，探索模型的协同训练方法，提升模型对整体样本空间的学习能力和多样性样本的生成能力。

3) 评估所提方法对不同网络环境中异常样本的检测性能，通过在SDN数据集和非SDN数据集上的实验，以及与传统集中和分布式检测方法的比较，验证所提方法的有效性和泛化能力。

1 相关研究工作

异常流量检测是通过一定的检测方法对网络中的流量数据进行分析和判断，并将非正常网络模式的数据判定为异常流量的过程。到目前，国内外研究人员提出许多针对SDN中异常流量的检测方法，按实现方式可分为集中式的异常流量检测和分布式的异常流量检测。

1.1 集中式的异常流量检测

集中式的异常流量检测方法通常由嵌入在服务器的检测组件集中完成流量数据的分析和异常判定工作。这种检测方法主要是将传统的基于统计分析、信息论、聚类分析，以及较新的基于机器学习和深度学习等检测方法迁移到SDN环境中实现的，是早期的SDN异常流量检测研究中的常用思路。

Bavani等[14]提出基于数据包统计信息的SDN中异常流量的检测方法，通过评估捕获到的原始数据包大小、持续时间，以及比对数据包的头部信息，对数据包的差异行为进行综合分析，从而判断SDN中的异常流量。然而，基于统计分析的检测方法只对特征明显的攻击行为有效，并且可识别的异常类型较少，不具备学习性，难以推广和拓展。周启钊等[15]则将交换机端口的统计量与流量差的信息熵结合，构建由各个关键特征弱分类器叠加形成的基于Boosting算法的增强型分类器，从而实现SDN中控制平面洪泛攻击的分类检测。这种基于信息论的检测方法只适合检测密度大的异常样本，对于稀疏异常则检测灵敏度较低并难以对异常的根因进行定位。Zolotukhin等[16]将K均值（K-means）、模糊C均值（fuzzy C-means）和自组织映射（SOM, self-organizing map）等聚类算法通过训练，集成为检测代理，根据计算出不同进程发送数据包的最大数量来定义判定阈值，然后利用质心聚类的方法来检测SDN数据包有效载荷中的异常数据。然而，聚类算法需要提前知道数据样本的种类数量才能更好地确定聚类中心，并且对具有不同异常密度的数据集适应性较差。

基于机器学习的检测方法在数据处理分析领域取得了比传统方法更加优异的成绩，在SDN中进行流量异常检测也得到推广应用。Tayfour等[17]将投票（voting）机制与朴素贝叶斯、K近邻、决策树和极度随机树等4个典型的基于机器学习的检测方法集成为V-NKDE分类器，利用分类器将提取到的SDN流量特征数据进行分类后，再利用投票机制确定最后结果，这种组合可以平衡各分类器自身弱点，降低误报率和过拟合风险。Satheesh等[18]提出一种利用机器学习确定优先级的异常检测模型，该模型通过分析数据包的详细信息从而对数据包进行分类整理，并利用SDN控制器调整流的转发规则来阻止恶意信息流。Sebbar等[19]提出基于随机森林算法的安全模型，在SDN中预先建立安全策略和生存时间（TTL, time to live）延迟标准，通过侦测上下文选择节点状态以识别中间人攻击，并将任何超过延迟标准的连接请求判定为攻击。基于机器学习的检测算法虽然在异常流量检测中取得一定成绩，但需要依赖特征工程支持，检测效果受工程人员的专业素养和技术水平制约。

深度学习是基于神经网络算法的机器学习，属于机器学习的一个分支，相对传统的最大似然模型具有更高的准确度，在SDN异常流量检测研究中有着诸多案例。卷积神经网络（CNN, convolutional neural network）、循环神经网络（RNN, recurrent neural network）是典型的监督学习模型，它们通过学习数据的分离边界实现对样本的分类。Wang等[20]提出将小波变换与CNN相结合的特征提取模型，并利用该模型提取SDN数据集样本中的多尺度潜在特征，最后使用分类器完成对异常样本的识别。Sri等[21]利用双向长短期记忆网络（LSTM, long short term memory network）提取SDN物联网流量数据的时间关联性特征，在二分类和多

分类的攻击检测中取得了较好效果。自编码（AE, autoencoder）网络和GAN是典型的无监督学习模型，它们专注于对数据内在特征的理解和表示。Yaser等[22]将LSTM和AE网络结合，构建基于深度堆栈自编码器的异常流量检测模型，利用当前自编码器的隐藏层作为后面自编码器的输入层，逐层提取数据的高阶抽象特征，以提升对原始样本的表征能力，从而提高异常检测准确率。Novaes等[23]在SDN中开发了一种基于对抗性训练的检测和预防系统，该系统利用GAN框架来识别DDoS攻击，并通过对抗训练来降低神经网络对对抗性攻击的敏感度，从而使异常检测系统能够更加准确地检测出攻击流量。此外还有介于监督学习与无监督学习之间的半监督学习，例如：Wang等[24]利用半监督生成对抗网络模型ByteSGAN对SDN边缘网关中的加密流量进行细粒度分类，他们首先将数据流转化为标准的字节向量，对于长度大于标准字节向量的数据包从尾部截断，对于长度小于标准字节向量的数据包则用“0”填充；然后利用标准字节向量对ByteSGAN判别器和生成器进行交叉训练，可在训练样本量较少的情况下提升检测的准确率。

从相关研究结果可以看出，对小规模的SDN异常流量检测而言，集中式的异常流量检测方法尚存在一些优势，但当面对大规模SDN异常流量检测任务时，往往存在检测任务超出单节点运算和存储能力的风险，容易引发单点故障问题。另外，在检测期间，数据的采样点需要不停地向检测点传输采样数据，这将大量消耗SDN带宽，严重时可能引发网络拥塞。将集中式的异常流量检测方法应用在大规模SDN的异常检测中，不仅会对SDN正常运行造成一定影响，甚至可能引发网络瘫痪。

1.2　分布式的异常流量检测

鉴于集中式的异常流量检测方法无法满足大规模SDN的检测需要，研究人员提出分布式的异常流量检测方法，根据实现方法可分为基于数据分布式的检测方法和基于模型分布式的检测方法。

1) 基于数据分布式的检测方法是指在不同的设备节点上使用相同检测模型处理数据集的不同子集，实现过程通常需要借助并行计算工具辅助完成。例如：Samaan等[25]使用Spark作为大数据工具创建分布式计算框架，将SDN中DDoS攻击检测任务分配给多台主机并行执行，以克服集中式的异常流量检测方法算力不足的问题。Patil等[26]提出基于Hadoop的分布式协作架构E-Had来检测DDoS攻击，通过将检测所需的计算和存储开销分配到多个映射器和缩减器中，提高处理流量数据的效率。Shukla等[27]针对基于物联网中的DDoS攻击，提出一种基于大规模分布式流计算平台的分布式检测方法。利用Hadoop集群和高度可扩展的H2O.ai机器学习平台创建5个分布式检测模型，并将模型在Apache流处理框架上部署形成分布式检测模型；另外，还将每个网络流中区分度较高的特征以及检测结果保存在Hadoop分布式文件系统（HDFS）中，从而实现对模型的实时更新。Kaur等[28]提出基于Kafka流的SDN中DDoS攻击检测模型，它是在双节点Apache Hadoop集群上使用机器学习技术实现的。2个节点的流量捕获模块对采集的网络流量数据进行特征提取，并将重要的网络特征发布到Kafka上；异常检测模块从Kafka上获取一半的网络流特征数据，对其进行分类后，最后再将检测结果发送到Kafka上。Ezeh等[29]提出基于多生成器和多判别器的GAN结构的高效异常检测（EGBAD, efficient GAN-based anomaly detection）模型，该模型利用多个控制器上的GAN结构，对样本数据进行异常检测，最后由中心服务器采用投票机制汇总每个GAN检测结果，以提升检测的准确性。

2) 基于模型分布式的检测方法则是将深度学习模型分割部署到多个设备上运行。例如：Parra等[30]提出将CNN的最后隐藏层嵌入后端服务器LSTM输入层的分布式检测模型（CNN-LSTM），该模型利用客户端上的CNN检测物联网中的URL攻击，同时将提取的特征数据传输到端服务器的LSTM上检测网络中的其他攻击类型。类似地，Feng等[31]提出基于SDN的分层分布式控制平面架构的攻击检测方案，该方案采用两层检测框架实现多域场景下DDoS攻击的协作检测。第一层检测方法采用基于信息熵算法的粗粒度异常检测方法，用于早期检测攻击路径上的DDoS攻击迹象；第二层检测方法采用基于混合深度学习模型的DCNN-LSTM算法，对可疑流量进行时空维度的细粒度DDoS攻击检测。肖警续等[32]提出基于SDN的物联网边缘节点间数据流零信任管理方法，将SDN应用到物联网边缘节点间的数据传输过程，通过交换

节点对数据的转发验证实现对数据篡改、转发路径异常和恶意丢弃等异常行为检测，并对恶意交换节点进行定位。陈何雄等[33]提出基于联邦学习的SDN异常流量协同检测技术，利用SDN的拓扑和流量特性构建出多检测节点的协同架构。该架构通过信息熵方法提取流量特征，并从相对熵角度分析检测节点间的流量关联度，利用参数聚合权重优化算法，根据流量关联度动态调整各检测节点在模型训练中的参数权重，从而提高检测模型的准确性。Shu等[34]使用有限的样本数据为整个车载自组织网络联合训练出可以单独检测常规网络流量和恶意网络流量的SDN控制器，并将控制器副本部署在其他控制器上。使各控制器能够完成所属子网的异常流量检测工作，从而减少通信和计算开销；然而，受训练样本规模和种类的限制，这种由单节点训练出来的检测模型会使整个SDN检测系统的泛化能力比较弱。

通过调研发现，当前基于数据分布式的异常流量检测方法，虽然缓解了集中检测时对检测节点的运算压力，但需要额外部署并行运算框架，会消耗SDN的存储空间，且在检测期间，各节点与控制台也存在着大量的数据交互，挤占了共享链路带宽。另外，基于模型分布式的检测方法，大都采用层次化设计，各模块间需要相互协调才能完成检测任务。然而，模块的复杂程度不同、算法本身的差异，加上检测节点性能也存在差距，使这种基于模型分布式的检测方法很难做到完美同步，检测模型的时效性不得不遵循“木桶效应”。另外，模块之间传输的检测数据也会消耗一定的网络资源。

针对当前基于分布式的异常流量检测方法存在的不足，本文提出一种基于深度学习的SDN异常流量分布式检测方法，该方法在模型训练阶段采用“一对多”的协同训练模式；在检测阶段，则由部署在各控制器上检测代理独立地完成所属子网的异常流量检测任务，从而最大限度地减小中心节点的运算开销和传输链路上的带宽消耗。

2　基于D-VAE-WGAN的分布式检测模型构建与运用

本文提出面向大规模SDN环境下异常流量的分布式检测主要依托基于D-VAE-WGAN的分布式检测模型（以下简称：D-VAE-WGAE模型）实现，它由部署在云端服务器的判别器与部署在各控制器的生成器组成，拓扑结构如图2所示。D-VAE-WGAN的分布式检测模型采取“集中训练、分布检测”的2阶段运行模式。在训练阶段，模型为“一对多”的模式设计，即一个云端服务器上的判别器对应多个控制器中的生成器，它们利用正常网络流量样本集中完成训练。训练完成后，云端服务器将训练结果的镜像副本发送到各控制器，在各控

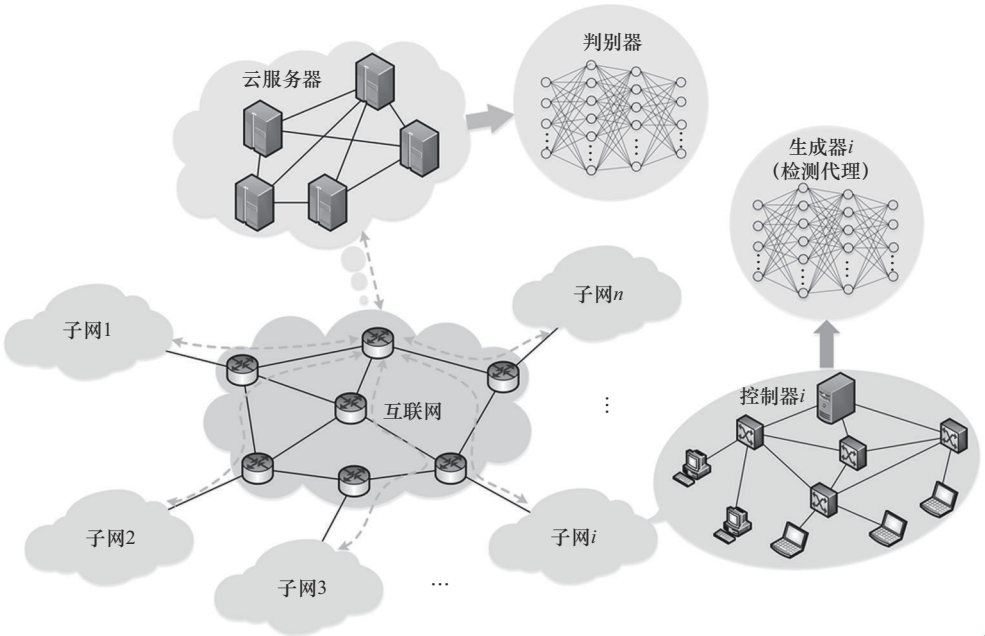


图2　基于D-VAE-WGAN的分布式检测模型的拓扑结构

制器上建立具有独立完成异常检测能力的检测代理。在检测阶段，检测代理对子网中流量数据进行深度分析，发现并识别其中的异常流量。这种集中与分散相结合的设计方式，既可全面监测SDN的状态信息，实现对SDN网络的整体防护，又可避免集中运算给网络设备带来局部高压。

2.1　D-VAE-WGAN模型架构

GAN最早由Goodfellow根据博弈论思想提出，它包括判别和生成器两部分，判别器和生成器又可以是CNN、RNN或者其他结构的神经网络，图3为GAN的基本结构。

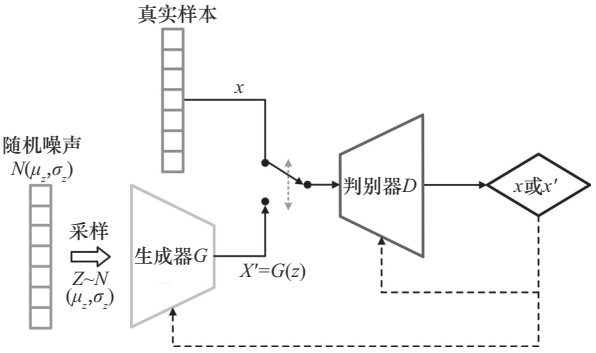


图3　GAN的基本结构

生成器将输入的随机噪声样本重构生成新的样本；判别器将对输入的样本进行判定，判断其来自真实样本集还是由生成器生成的样本，并将判断结果反馈给生成器；生成器根据反馈的结果调整参数再生成新的样本，判别器则进行再判定、再反馈，如此反复迭代多次。因此，GAN的训练过程其实就是判别器和生成器相互博弈的过程，通常情况下先利用真实样本对判别器进行训练；生成器尽力生成近似样本去欺骗判别器，并通过不断地迭代完善，直到生成器学习到真实样本的数据分布生成逼真样本，使判别器无法区分数据的来源，最终两者达到纳什均衡。GAN的优化目标可以描述为一个求极小值和极大值的问题。

$$\min_G \max_D V(D,G)=\mathbb{E}_{x\sim P_r}[\log(D(x))]+\mathbb{E}_{z\sim P_g}[\log(1-D(G(z)))] \tag{1}$$

其中，x为真实数据样本，P_r表示真实样本分布，z为来自随机分布P_g的噪声数据，$G(z)$为生成数据，$D(x)$为样本x来自真实样本空间的概率。可以看出，GAN无须先验知识就能学习到数据分布，突破了对数据标注和正负样本平衡性要求的限制。

然而，普通的GAN如果没有合适的编码器指导训练，将会出现模型收敛缓慢，训练困难等问题；另外，GAN使用库尔贝克-莱布勒（KL，Kullback-Leibler）散度作为数据分布的差异性度量，也会导致模型在训练时出现“梯度消失或爆炸”、收敛困难无法完成训练的问题，以及生成样本模式单一的“模式崩溃”问题。为克服普通GAN自身结构的不足，文献[35]将VAE与GAN相结合构造出VAE-WGAN架构，利用VAE作为GAN的编码器辅助GAN完成训练；并引入Wasserstein距离作为相似性度量，避免发生“模式崩溃”问题。本文在VAE-WGAN基础上，根据大规模SDN拓扑特点，提出D-VAE-WGAN模型架构，如图4所示。可以看出，每个分布式控制器包含一个完整的VAE（包含编码器和解码器）和一个GAN的判别器。云端服务器上只有一个GAN的判别器，它与各控制器上的生成器组成“一对多”的生成对抗网络结构。

2.2　基于Wasserstein距离的联合优化及基于标准差约束算子的正则化方法

1) 基于Wasserstein距离的联合优化

在SDN控制器内部，VAE由编码器G_e和解码器G_d组成（G_e和G_d在2个GAN中作为生成器），主要是学习输入样本x的特征，并通过编码和解码的方式得到重构样本$G_d(G_e(x))$。使用周期一致性损失函数训练编码器G_e和解码器G_d，防止它们作为GAN生成器的时候产生矛盾，目标函数为

$$\min_{\{G_e,G_d\}} V_{L_2}(G_e,G_d)=\mathbb{E}_{x\sim P_X}\left\|x-G_d(G_e(x))\right\|_2 \tag{2}$$

另外，VAE的编码器G_e作为生成器和判别器D_e组成一个GAN，对于输入$x\sim P_X$，VAE会产生中间输出$G_e(x)\sim Q_Z$；引入高斯分布P_Z作为约束，利用随机变量z找到条件分布$Q(Z|X)$，使z的边界$Q_Z(Z):=\mathbb{E}_{X\sim P_X}[Q(Z|X)]$，即$Q_Z=\int Q(Z|X)\mathrm{d}P_X$，通过调整$G_e$的参数使$G_e(x)$服从$Q_Z$分布并与先验分布$P_Z$之间的距离不断减小，以实现训练生成器$G_e$的目标，利用Wasserstein距离作为$G_e$与$D_e$组成的WGAN-*e*的目标函数，可表示为

$$\min_{G_e}\max_{\|D_e\|\leqslant 1} V(D_e,G_e)=\mathbb{E}_{z\sim P_Z}[D_e(z)]-\mathbb{E}_{x\sim P_X}[D_e(G_e(x))] \tag{3}$$

在云端服务器，判别器D_d与VAE解码器G_d作为生成器，组成另一个WGAN-*d*结构。在WGAN-*d*中仅使用范数距离并不能很准确地描述重构数据与

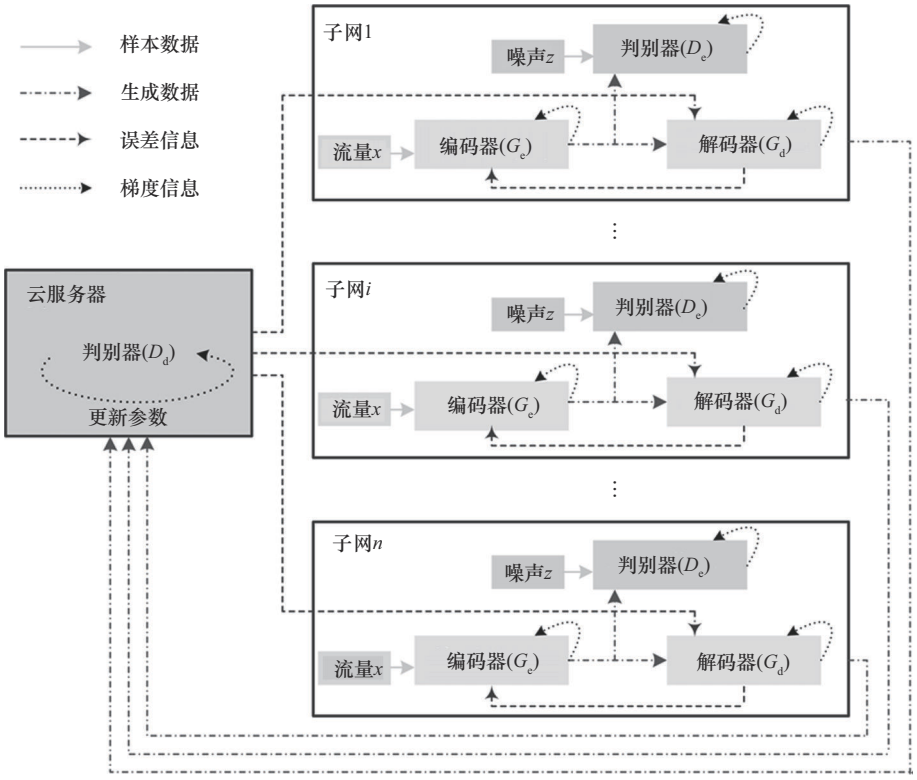


图4 D-VAE-WGAN模型架构

输入数据的相似性，为了让变分自编码器的重构输出更接近于输入数据的真实分布，提出利用判别器D_d进一步约束重构数据与输入数据的差异，WGAN-*d*同样使用Wasserstein距离作为目标函数

$$\min_{G_d}\max_{\|D_d\|\leqslant 1} V(D_d,G_d)=\mathbb{E}_{x\sim P_X}[D_d(x)]-\mathbb{E}_{x\sim P_X}[D_d(G_d(G_e(x)))] \tag{4}$$

GAN判别器和生成器的训练是异步的，因此每个VAE-WGAN模型分别对应了3个异步的训练过程，它们都有各自的损失函数及优化器。由于GAN的判别器在训练阶段，只涉及自身，因此可以直接用$z\sim P_Z$中的样本作为输入来训练D_e；用$x\sim P_X$中的样本作为输入训练D_d。而2个GAN的生成器G_e和G_d还作为VAE的编码器和解码器，因此在训练G_e、G_d和VAE时需同时兼顾式(2)~式(4)，利用它们3个目标函数的加权和作为最终的目标函数。

$$\min_{\{G_e,G_d\}}\max_{\{\|D_e\|\leqslant 1,\|D_d\|\leqslant 1\}} \lambda V_{L_2}(G_e,G_d)+\gamma V(D_e,G_e)+\mu V(D_d,G_d) \tag{5}$$

其中，λ、γ、μ为各目标函数的权重，并且它们的和为1，本文中它们取值分别为0.4、0.3、0.3。

2) 基于标准差约束算子的正则化方法

传统的参数正则化方法只关注单个特征的权重值，而没有考虑特征之间的内在联系。考虑到网络流量属性特征并不是孤立存在的，这些神经元权重元素之间存在着某种关联性，为了更好地利用特征权重之间的关联信息。根据多层神经网络构建权重矩阵

$$\boldsymbol{\omega}=\begin{bmatrix}\omega_{11} & \omega_{12} & \omega_{13} & \cdots & \omega_{1m}\\ \omega_{21} & \omega_{22} & \omega_{23} & \cdots & \omega_{2m}\\ \omega_{31} & \omega_{32} & \omega_{33} & \cdots & \omega_{3m}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \omega_{n1} & \omega_{n2} & \omega_{n3} & \cdots & \omega_{nm}\end{bmatrix} \tag{6}$$

其中，n为多层神经网络层数，m是所有神经网络层中神经元的最大数，ω_{ij}为第i层第j个神经元的权重值；对应位置无神经元的，权重值为0。根据标准差计算方式，设计出基于标准差（SD，standard deviation）的约束算子

$$\sigma(\boldsymbol{\omega})=\sqrt{\frac{1}{nm}\left\{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\omega_{ij}-\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\omega_{ij}\right)^2\right\}} \tag{7}$$

利用标准差约束算子作为正则化项，构建损失函数

$$\text{Loss}=f_w(x)+\alpha\sigma(\boldsymbol{\omega}) \tag{8}$$

其中，$f_w(x)$为误差函数，α用于控制约束算子的相对重要性，此正则化方法可避免模型训练过拟合风险，并且有利于挖掘流量数据间的潜在关联性，为数据样本分类提供更加丰富的特征信息。

2.3　D-VAE-WGAN模型协同训练方法

D-VAE-WGAN模型的VAE中编码器和解码器采用不含池化层的多层卷积网络，编码器输入层和解码器输出层的神经元个数与待检测数据维度相同。判别器D_e和D_d为不含Sigmoid层的一维卷积神经网络结构。模型的超参数根据经验法则和实验结果调整得到，具体如表1所示，其中平滑常数是RMSProp优化算法使用，类似于梯度动量参数的功能，主要是为了使参数空间变得更为平缓，有利于提升模型的收敛速度。

表1　D-VAE-WGAN模型的相关超参数

超参数	取值
总体训练次数	100
批大小	256
学习率	0.00 1
平滑常数	0.9
优化器	RMSProp
周期迭代次数k	5

使用正常流量样本对云端服务器与各SDN控制器的D-VAE-WGAN模型进行交互协同训练，包括多个迭代步骤，具体过程可描述如下。

1) 第i个SDN控制器利用输入样本x_i和高斯噪声z_i，完成D_e自身参数ω_{D_e}的更新；VAE生成样本数据$G_d(G_e(x_i))$，并发送至位于云端服务器端的判别器D_d。

2) 判别器D_d根据接收到控制器发送的输入样本数据x_i，完成自身参数ω_{D_d}的更新。

3) 判别器D_d根据接收到控制器发送的生成数据$G_d(G_e(x_i))$，计算生成误差，并将结果反馈给第i个控制器。

4) 控制器根据反馈回的误差信息更新G_e、G_d的参数ω_{G_e}和ω_{G_d}。

5) 在完成一个k轮的周期性迭代之后，通过随机选择方式，与第j个控制器交换参数ω_{G_e}和ω_{G_d}。

6) 重复步骤1)～步骤5)，直到完成总体迭代次数要求。

使用随机交换参数的训练方式可以进一步避免对控制器训练的过拟合问题，同时还可共享其他控制器的训练成果，使每个控制器学习到SDN中其他控制器子网中流量数据的分布，从而形成对大规模SDN的整体防御能力。

2.4　基于检测代理的异常检测

模型完成训练后，云端服务器将训练好的D_d的副本发送到各子网控制器中，与控制器上原有的G_e、G_d和D_e共同构成检测代理，该代理可以独立完成本子网络的流量监测任务。由于使用正常流量建模，检测代理对输入的正常流量样本可以进行很好地重构，而对于异常样本的重构则会产生较大的重构误差和判别误差，因此将样本的误差值定义为重构误差和判别误差的加权组合，即

$$C(x)=\alpha Z_{\mathrm{Re}}(x)+(1-\alpha)Z_{D_d}(x) \tag{9}$$

其中，$Z_{\mathrm{Re}}(x)$为重构误差，它是输入样本x与重构样本$\hat{x}$在对应维度特征值差异的总和，计算方法为

$$Z_{\mathrm{Re}}(x)=\|x-\hat{x}\|_1=\sum_{i=1}^{m}|x_i-\hat{x}_i| \tag{10}$$

$Z_{D_d}(x)$则是判别误差，它是判别器认为真实输入样本和重构样本之间的差异，仍使用Wasserstein距离表示，即

$$Z_{D_d}(x)=\inf_{\gamma\sim\prod(P_r,P_g)}\mathbb{E}_{(x,\hat{x})\sim\gamma}[\|x-\hat{x}\|] \tag{11}$$

另外，$\alpha\in(0,1)$为控制两项相对重要性的参数，在没有特别要求的情况下，认为重构误差与判别误差在异常判定中发挥的作用同等重要，因此，本文中α的取值为0.5。

计算得出样本的误差值，使用阈值法判断该样本是否属于异常样本。阈值的设置按照“三西格玛准则”进行，通过计算训练集所有正常样本误差值的均值再加3个标准差求得，即

$$C_{\mathrm{threshold}}=\bar{C}+3\sigma_C \tag{12}$$

在异常检测时，控制器上的检测代理对子网中所有的流量样本进行检测，当检测到某个样本的误差值大于判断的阈值时，即可将判定该样本为异常样本，即

$$\mathrm{Attribute}(x)=\begin{cases}\text{异常}, C(x)>C_{\mathrm{threshold}}\\ \text{正常}, \text{其他}\end{cases} \tag{13}$$

2.5　数据设计

使用InSDN数据集作为实验数据，它是一款专门针对SDN环境进行流量分析实验的数据集[36]。根据流量类型和目标主机，InSDN数据集可分为3组：第一组仅包括正常流量，由当前常用的HTTPS、HTTP、Email、DNS等协议产生，共68 424条记录。第二组是以Metasploitable2服务器为目标的攻击流量，由

DoS、DDoS、蛮力攻击（Brute force）、探测（Probe）和远程非授权访问（U2L，remote to local）5种攻击产生，共138 722条记录。第三组为OVS服务器内部的攻击流量，它包含僵尸网络（BotNet）、DoS、DDoS、Brute force、Probe和Web-Attack这6种攻击产生的流量数据，共136 743条记录。为减少异常检测时的运算开销，本文从.csv格式数据集文件的80多个特征中选择包括传输协议，流的持续时间、空闲时间、流字节率、流包率、流包间隔，以及数据包的长度、包头大小、包速率、传输时间、传输间隔等48个相关特征作为模型训练和检测的实验数据[37]，InSDN数据集特征子集如表2所示。

表2 InSDN数据集特征子集

编号	特征名称	编号	特征名称
1	Protocol	25	Fwd IAT Total
2	Flow duration	26	Bwd IAT Min
3	total Fwd Packet	27	Bwd IAT Max
4	total Bwd packets	28	Bwd IAT Mean
5	total Length of Fwd Packet	29	Bwd IAT Std
6	total Length of Bwd Packet	30	Bwd IAT Total
7	Fwd Packet Length Min	31	Fwd Header Length
8	Fwd Packet Length Max	32	Bwd Header Length
9	Fwd Packet Length Mean	33	FWD Packets/s
10	Fwd Packet Length Std	34	Bwd Packets/s
11	Bwd Packet Length Min	35	Packet Length Min
12	Bwd Packet Length Max	36	Packet Length Max
13	Bwd Packet Length Mean	37	Packet Length Mean
14	Bwd Packet Length Std	38	Packet Length Std
15	Flow Bytes/s	39	Packet Length Variance
16	Flow Packets/s	40	Average Packet Size
17	Flow IAT Mean	41	Active Min
18	Flow IAT Std	42	Active Mean
19	Flow IAT Max	43	Active Max
20	Flow IAT Min	44	Active Std
21	Fwd IAT Min	45	Idle Min
22	Fwd IAT Max	46	Idle Mean
23	Fwd IAT Mean	47	Idle Max
24	Fwd IAT Std	48	Idle Std

由表2可以看出，没有选择源IP、目的IP、源端口号和目的端口号作为特征集的元素，这样做的目的是增加检测模型的泛化能力。因为，不同网络环境下的IP及端口号通常存在较大差异，如果把它们作为特征数据来训练模型，势必将限制模型的应用范围，不利于模型泛化能力的生成。

为了消除特征数据量纲不一致对检测结果带来的不利影响，采用Min-Max标准化方法，对特征值按照一定比例进行缩放，使它们落在一定的区域，计算方法为

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{14}$$

其中，x为特征的原始值，$x_{\min}$、$x_{\max}$分别为该特征值的最小值和最大值。经标准化后所有的特征数据都映射在[0,1]。最后，按7:3的比例对各类样本进行划分，利用70%的正常样本组成无监督学习的训练集A，再将这70%的正常样本和异常样本组成监督学习的训练集B，剩下的正常样本和异常样本组成测试集A。另外，从测试集A中每种类型的样本中随机抽取50%组成测试集B1，剩下的样本组成测试集B2。考虑到U2R样本数量过于稀少（仅有17个），不适合参加训练和测试，故将其剔除，则InSDN数据集划分后的样本分布如表3所示。

表3 InSDN数据集划分后的样本分布

样本类型	原始集	训练集A	训练集B	测试集A	测试集B1	测试集B2
DDoS	121 942	0	85 359	36 583	18 292	18 291
DoS	53 616	0	37 531	16 085	8 043	8 042
Probe	98 129	0	68 690	29 439	14 720	14 719
Brute Force	1 405	0	983	422	211	211
Web-Attack	192	0	134	58	29	29
BotNet	164	0	115	49	25	24
U2L	17	0	0	0	0	0
Normal	68 424	47 897	47 897	20 527	10 264	10 263
总计	343 889	47 897	240 709	103 163	51 584	51 579

3 实验与结果分析

3.1 评估指标

为评估检测方法对网络攻击的检测性能，令异常样本为正例、正常样本为负例，采用准确率

（Accuracy）、精确率（Precision）、召回率（Recall）、F1值以及误报率（FPR）等5项检测指标作为评价检测代理性能的参考，它们的计算方法如下。

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{15}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

$$\text{F1值} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{19}$$

其中，TP、FN、FP、TN的含义可由表4中的关系矩阵表示。

表4　真实值与预测值的关系矩阵

真实值	预测为正例	预测为负例
真实为正例	TP	FN
真实为负例	FP	TN

3.2　实验设置

本文实验的服务器配置为Core i9-12900F处理器、NVIDIA RTX3090显卡、128 GB内存、CUDA11.2、Pytorch1.8。另外，使用5台不同型号的主机运行POX程序作为控制器；每部控制器主机运行MinNet程序模拟4台交换机，每台交换机下连接4台主机，构建出SDN子网络；控制器与服务器通过路由器和交换机连接，构建主从结构的分布式SDN实验环境。服务器与各控制器的配置信息如表5所示。

表5　服务器与各控制器的配置信息

设备名称	处理器型号（频率）	内存容量（频率）
服务器	Intel Core i9-12900F (2.4 GHz)	128 GB(4 800 MHz)
控制器1	Intel Core i7-8750 (2.21 GHz)	32 GB(2 666 MHz)
控制器2	Intel Core i7-12700H (2.3 GHz)	32 GB(4 800 MHz)
控制器3	Intel Core i7-8750 (2.21 GHz)	16 GB(2 666 MHz)
控制器4	AMD Ryzen R7-4800H (2.9 GHz)	16 GB(3 200 MHz)
控制器5	Intel Core i7-8565U (1.8 GHz)	16 GB(2 133 MHz)

3.3　结果分析

3.3.1　独立检测实验与分析

本阶段主要对单个检测代理的异常检测能力进行评估。利用训练集A对D-VAE-WGAN模型进行完全训练后，在各控制器上完成检测代理的构建。利用检测代理依次对测试集A进行独立异常流量检测实验。检测期间，云端服务器和其他检测代理暂停工作。表6展示了5个检测代理在测试集A上独立进行异常检测的实验结果。

表6　检测代理对测试集A的独立异常检测结果

检测代理	准确率	精确率	召回率	F1值	误报率
代理1	97.75%	98.76%	98.43%	98.59%	4.97%
代理2	97.75%	98.64%	98.55%	98.59%	5.46%
代理3	97.81%	**98.78%**	98.48%	98.63%	**4.88%**
代理4	97.81%	98.67%	98.60%	98.63%	5.36%
代理5	**97.93%**	98.61%	**98.81%**	**98.71%**	5.61%

由表6可以看出，各检测代理对测试集A检测的准确率都在97.75%以上，最高为97.93%；对异常样本检测精确率最高的为检测代理3达到98.78%、最低的为检测代理5也取得98.61%；对异常样本的召回率最高的为检测代理5达到98.81%、最低的为检测代理1取得98.43%；几个检测代理的误报率均为5%左右。实验结果表明，各检测代理均可独立地完成对异常样本的准确识别，并且误报率也在可接受范围内。另外，对比各检测代理的结果可以看出，虽然它们的性能有稍许差异，但这种差距并不明显，这是由于D-VAE-WGAN模型在训练时采用了交换参数后再训练的方法，使各检测代理对数据的偏好差异不大，故而表现出相近的检测性能。

3.3.2　联合检测实验与分析

本阶段主要讨论多个检测代理共同工作时的检测效果，为综合评估所提方法对整个SDN异常流量的检测性能，云端服务器也对各检测代理的检测结果进行统计。考虑到各检测代理所在主机在性能上存在差异，检测过程会出现不同步的情况，为便于统计分析，对测试集A的所有样本进行统一且唯一地编号，且编号不作为异常检测的特征字段。另外，当有检测代理对同一编号样本的检测结果存在争议时，云端服务器采用“少数服从多数”的投票机制确定该样本类型。例如：有2个检测代理判定某个样本为“正常”，3个检测代理认为该样本为“异常”，则云端服务器判定该样本为异常样本，反之亦然。图5展示了各检测代理及服务器对测试集A的联合检测结果。

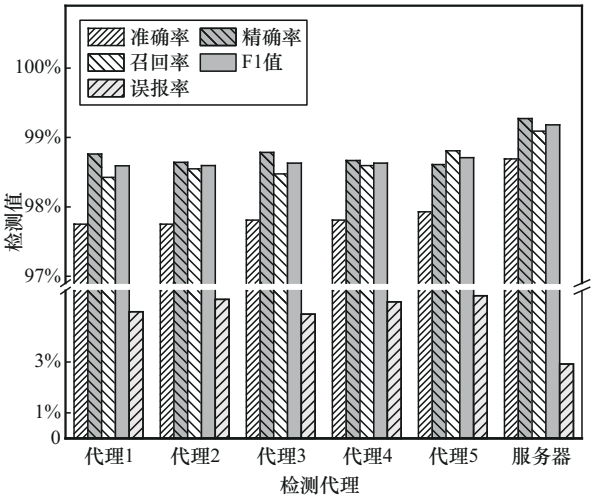


图5 各检测代理及服务器对测试集A的联合检测结果

几个检测代理在测试集A上的表现依旧良好，这说明，在大规模SDN中，各检测代理是独立执行检测任务的，子网间互不干扰。另外，服务器统计的检测指标明显要高于单个检测代理。这是由于服务器在汇总几个检测代理检测结果时，对于一些被检测代理错误分类的样本，经投票后得到纠正，因此服务器表现出的检测性能要明显高于单个检测代理。总体来说，D-VAE-WGAN模型训练出来的多个检测代理能够有效地完成所属子网的异常流量检测任务且总体表现良好。

3.3.3 检测耗时实验与分析

为了衡量检测代理对异常样本检测的时间效率，记录各检测代理在测试集A、测试集B1和测试集B2上的检测耗时。通常情况下，异常检测的耗时取决于神经网络的类型、模型复杂度、实验环境以及数据集大小等因素。本实验中各检测代理的模型结构相同，但被部署在不同性能的主机上；另外，3个测试集的样本量不同，测试集A的样本总量为103 163条记录、测试集B1的样本总量为51 584条记录、测试集B2的样本总量为51 579条记录。图6展示了各检测代理在不使用GPU加速的情况下对3个测试集的检测耗时。

由图6可以看出，检测代理2在3个数据集的检测耗时最少，分别为353.84 s、200.26 s和219.32 s；而检测代理5耗时最多，分别为466.86 s、312.32 s和290.66 s。检测时间上的差异主要由检测代理所在主机的性能决定，控制器2所在主机是几台主机中综合性能最好的，故而运算速度最快；而控制器5所在主机的综合性能较差，检测耗时最长。另外，从检测代理在3个数据集的检测耗时可以看出，虽然3个数据集样本量比为2:1:1，但在较大数据集的检测耗时并不是较小数据集检测耗时的2倍。这说明检测代理具备处理大体量样本数据的能力且检测耗时相对更短。此外，检测代理对单个样本的检测时间为2~5 ms，具备执行实时在线检测任务的潜力。

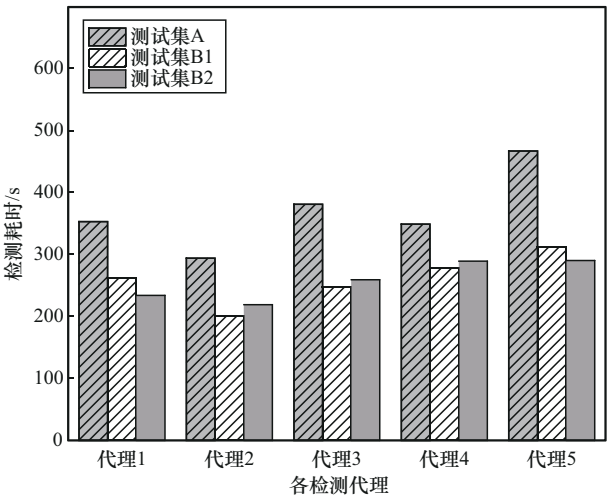


图6 检测代理在各测试集上的检测耗时

3.3.4 在不同数据集上的检测实验

为验证所提方法的泛化能力，在流行的CIC-IDS2017、CSE-CIC-IDS2018、CIC-DDoS2019、UNSW-NB15等公开网络流量数据集上对检测代理性能进行评估。CIC-IDS2017，CSE-CIC-IDS2018和CIC-DDoS2019由加拿大网络安全研究所提供，按2.5节的方法选择48个特征数据；UNSW-NB15由澳大利亚网络安全中心提供，选用除标签外的48个特征作为实验数据。从它们各自的原始数据集中抽取出一定量的正常样本并按7:3的比例划分，将70%的正常样本作为训练集，再次抽取一定量的异常样本与另外30%的正常样本组成不同异常样本占比的测试集，具体信息如表7所示。

首先，从4个数据集中选择一个，利用它的训练集对D-VAE-WGAN模型进行完全训练，并完成各个子网检测代理的构建。再利用该数据集的测试集对各检测代理的检测性能进行测试，并记录检测结果。然后利用其他数据集进行模型训练、检测代理构建、异常检测实施、记录检测结果，直到几个数据集都完成测试。图7展示了各检测代理对4个数据集中异常样本的检测性能。

表7　　各网络流量数据集的相关信息

数据集名称	异常样本类型	原始特征数	使用特征数	训练集（正常样本）	测试集		
					正常样本	异常样本	异常样本占比
UNSW-NB15	DoS, Fuzzers, Reconnaissance, Port Scans, Worms, Generic, Exploits, Shellcode, Backdoors	49	48	65 100	27 900	164 673	85.51 %
CIC-IDS2017	Brute force, Portscan, Botnet, DoS, DDoS, Web, Infiltration	83	48	291 425	124 896	288 923	69.82 %
CSE-CIC-IDS2018	Brute force, DoS, Web Attack, LOIC, Portscan, SlowHTTPTest, LOIC, DDoS, SQL Injection, Infiltration , Botnet	83	48	743 278	318 548	140 635	30.63 %
CIC-DDoS2019	MSSQL, NetBIOS, NTP, SNMP, SSDP, SYN, TFTP, DNS, UDP, UDP-Lag, Web DDoS, LDAP	80+	48	44 600	19 100	20 013	51.17 %

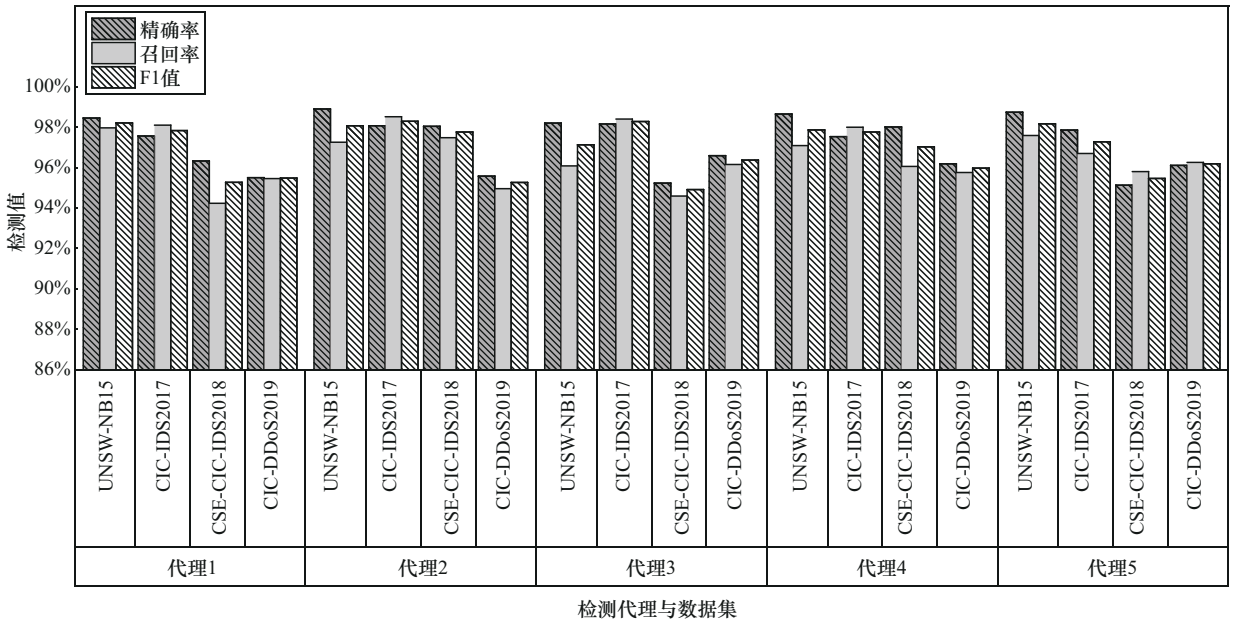


图7　检测代理在各数据集的检测结果

从图7中可以看出，各检测代理对异常样本检测的精确率最高为98.89%、最低为95.13%；召回率最高为98.51%、最低为94.33%。这说明基于D-VAE-WGAN模型生成的检测代理对异构网络中的流量数据也具有较强的判别能力。通过对比发现，多数检测代理在UNSW-NB15的精确率指标都比较高，原因在于UNSW-NB15数据集中异常样本占比相对较高，为85.51%，这样即使有一定量的正常样本被误报，对精确率指标的影响也不明显，所以各检测代理在UNSW-NB15的精确率都比较高。另外，各检测代理在CSE-CIC-IDS2018上的检测指标相较于CIC-DDoS2019波动比较大，原因在于CSE-CIC-IDS2018数据集中异常样本占比相对稀少，精确率和召回率两项指标对异常样本的检出数量较为敏感；而CIC-DDoS2019中正常样本与异常样本数量相当，故而2项检测指标的变化相对平稳些。总体来说，检测代理在4个数据集上的检测效果都较为出色，说明D-VAE-WGAN模型对不同异常占比的网络数据也有着较好的适应性。

另外，D-VAE-WGAN模型仅使用正常流量样本即可完成训练，检测代理在训练阶段即使没有见过异常流量样本，在检测阶段也能很好地检测出异常样本，这说明，检测代理对于未知的异常样本具有很好的识别能力。

3.3.5　与其他检测方法的对比实验

为更加客观地评估D-VAE-WGAN模型对异常流量的检测能力，选取了几个比较有代表性的检测模型，利用InSDN数据集进行对比实验。这些模型包括：V-NKDE[17]、EGBAD[29]、CNN-LSTM[30]。

V-NKDE为集成了朴素贝叶斯、K近邻、决策树和极度随机树4个机器学习的检测模型，是典型的集中式的异常流量检测方法。EGBAD作为数据分布式检测模型的代表，与D-VAE-WGAN模型类似，它也是由多个GAN组成的架构，不同的是EGBAD采用“多对多”的关系，并且它的训练与检测都是在相同的网络模式下进行，实验中使用5个生成器和5个判别器的结构。CNN-LSTM为基于模型分布式的检测方法，客户端中的CNN在提取样本特征的同时也可完成对URL攻击的检测，本实验只利用到它的特征提取功能，为后端服务器中的LSTM提供特征数据，最终的异常检测任务仍在后端服务器完成。

由于V-NKDE、CNN-LSTM为监督学习模型，需要带标签的样本完成训练，因此我们利用训练集B对V-NKDE、CNN-LSTM这2个模型进行监督训练，对EGBAD则使用训练集A进行无监督训练。最后，使用测试集A对V-NKDE、CNN-LSTM和EGBAD完成性能测试。对于D-VAE-WGAN的检测实验则将测试集A按样本类别随机等分为5组，每一组输入一个检测代理中。由于各检测代理所接收的检测样本均不相同，不可使用投票机制汇总检测结果，因此云端服务器需对各检测代理的测试结果进行简单统计，各模型最终检测结果如表8所示。

从表8中可以看出，5个检测代理在每项检测指标中的表现都较为出色。其中，准确率、召回率和F1值最高的前3个都是检测代理；精确率和召回率最高的前3个中有2个是检测代理。根据最终的统计结果看，D-VAE-WGAN是几个检测模型中性能指标最好的，这说明D-VAE-WGAN在同等条件下训练出的检测模型较传统方法有着更加出色的异常流量检测能力。

在资源消耗方面，V-NKDE是集中式的异常流量检测方法，检测模型部署在服务器端；检测时，4个机器学习模块都要对所有的样本数据进行一次完整的检测，然后利用投票机制汇总出最终结果，因此整个检测过程中，V-NKDE的服务器要进行4倍测试集样本量的运算。CNN-LSTM采用的是两阶段的分布式的异常流量检测方法，先由各子网客户端的CNN对输入样本进行特征预提取，同时将提取的特征数据映射到后端服务器的LSTM再进行深度检测。此过程中，客户端会持续向后端服务器发送提取的特征数据，客户端与后端服务器之间的通信链路将被持续占用，而后端服务器也需要不断地处理来自各客户端发来的数据，因此，整个CNN-LSTM要进行2倍测试集样本量的运算，其中各客户端共处理1倍测试集，LSTM处理1倍测试集。EGBAD中的每个GAN都可以实现分布式检测，但设计人员为追求更高准确率，要求每个GAN不仅要检测本子网的数据，还需要检测其他子网的生成样本，最终结果取各子网结果的平均值，这相当于

表8 不同检测方法的实验结果对比

检测模型	准确率	精确率	召回率	F1值	误报率	检测范围	处理数据量	共享链路数据	服务器参与检测	检测耗时/s
V-NKDE	96.70%	98.67%	97.19%	97.93%	5.26%	全网	4×Test-A	样本数据	是	1793.84
CNN-LSTM	96.50%	97.84%	97.79%	97.82%	8.67%	全网	2×Test-A	特征数据	是	486.12
EGBAD	97.78%	**98.73%**	98.50%	98.61%	5.12%	全网	5×Test-A	生成样本	否	531.84
代理1	97.87%	98.79%	98.55%	98.67%	4.87%	子网	$\frac{1}{5}$×Test-A	无	否	84.19
代理2	97.74%	98.67%	98.51%	98.59%	5.36%	子网	$\frac{1}{5}$×Test-A	无	否	76.46
代理3	98.04%	98.80%	98.75%	98.78%	4.82%	子网	$\frac{1}{5}$×Test-A	无	否	98.26
代理4	97.92%	98.68%	98.72%	98.70%	5.31%	子网	$\frac{1}{5}$×Test-A	无	否	89.19
代理5	97.90%	98.72%	98.66%	98.69%	5.16%	子网	$\frac{1}{5}$×Test-A	无	否	118.72
D-VAE-WGAN	**97.89%**	**98.73%**	**98.64%**	**98.68%**	**5.11%**	全网	Test-A	检测结果	否	122.25

EGBAD中每个子网都要处理一个测试集，这种方法显然需要消耗更多的计算和存储资源，另外子网间也会有大量的生成样本数据交互。D-VAE-WGAN训练出的各检测代理可以独立地完成所属控制器子网的异常检测任务，云端服务器只负责记录和统计各子网的检测结果，并不参与检测运算，因此每个子网只需处理$\frac{1}{5}$的测试集样本。整个D-VAE-WGAN数据处理量为1倍测试集，相较于其他几个模型，运算压力和带宽消耗最小。因此D-VAE-WGAN模型也是几个检测模型中耗时最少的。

3.3.6　消融实验

本节主要采用“消融实验”的办法，探讨基于标准差约束算子的正则化方法以及随机交换参数的协同训练方法对模型检测的影响。

1) 基于标准差约束算子的正则化方法

为降低模型训练出现过拟合的风险，通常会在模型的损失函数中添加正则化项对参数进行约束，使模型更加平滑和稳定，常见的有L1正则化项、L2正则化项。为探究所提的基于标准差约束算子的正则化方法对模型训练及异常检测的影响，在其他条件保持不变的情况下，分别以L1和L2作为D-VAE-WGAN模型损失函数的正则化项，构建检测模型D-VAE-WGAN(L1)和D-VAE-WGAN(L2)，并使用相同的方法对它们进行训练和异常检测。图8显示了它们利用训练集A训练时，与原来D-VAE-WGAN模型损失函数变化情况的对比。另外，为便于区分，本节将原来基于标准差约束算子的D-VAE-WGAN表述为D-VAE-WGAN(SD)。

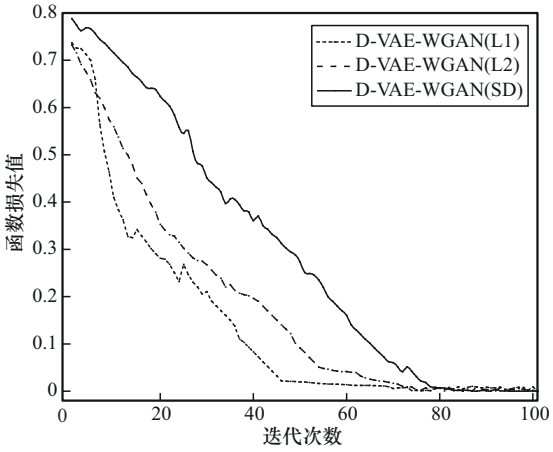


图8　不同正则化方法下模型训练过程对比

由图8可以看出，D-VAE-WGAN(SD)在训练过程中，损失误差起始阶段下降得比较平缓，不如L1正则化或者L2正则化收敛速率快，这是由于标准差约束算子使用的是权重矩阵的标准差，而非权重参数绝对值的和或者平方和，对损失函数值整体的变化影响相对较小，因此表现出平缓的变化形态。不过，随着训练持续D-VAE-WGAN(SD)也逐渐收敛到稳定状态。

按3.3.5节的检测方式评估D-VAE-WGAN(L1)和D-VAE-WGAN(L2)在测试集A上的检测结果，并与D-VAE-WGAN(SD)已取得的结果进行比较，如图9所示。

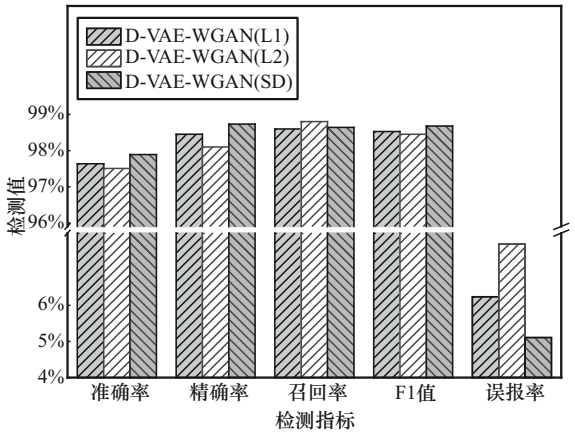


图9　不同正则化方法下模型检测结果对比

由图9可以看出，D-VAE-WGAN(SD)的检测准确率和精确率指标均为最高，分别为97.89%和98.73%，召回率为98.64%比D-VAE-WGAN(L2)的98.80%略低，但明显高于D-VAE-WGAN(L1)的召回率；另外，D-VAE-WGAN(SD)的F1值最高，且误报率最低。这说明基于标准差约束算子的正则化方法建立的检测模型比基于L1正则化项和L2正则化项的检测模型能从输入的流量数据中挖掘出更多的特征关联性信息，这些信息对提升模型的检测性能有着促进作用。

2) 协同训练方法

采用协同训练的目的是使模型通过交换训练参数，增强对数据分布的学习效果，并提升模型训练的效率。为客观评估这种训练方式对模型检测效果的影响，将其与非协同训练建立的模型以及仅使用部分训练集数据训练出的模型进行比较。为便于表述，本节将原来采用协同训练的D-VAE-WGAN表

述为D-VAE-WGAN(C)，将非协同训练建立的模型表述为D-VAE-WGAN(NC)；对使用部分训练集进行协同训练建立的模型表述为D-VAE-WGAN(C-sub)、非协同训练建立的检测模型表述称为D-VAE-WGAN(NC-sub)。

使用训练集A对D-VAE-WGAN(NC)进行完全训练，再将训练集A随机等分为5个子集，分别对D-VAE-WGAN(C-sub)和D-VAE-WGAN(NC-sub)的5个子网生成器进行完全训练，当它们的损失函数不再随训练次数的增加而明显减小时停止训练。将云端服务器的训练成果映射到各子网的控制器中构造检测代理，使用测试集A按3.3.2节的方法对检测代理进行异常流量检测实验，服务器统计检测结果，并与D-VAE-WGAN(C)的检测结果进行对比，结果如表9所示。

表9 协同训练模型与非协同训练模型的性能对比

模型	准确率	精确率	召回率	F1值	误报率
D-VAE-WGAN (NC)	96.58%	97.50%	98.25%	97.87%	10.13%
D-VAE-WGAN (NC-sub)	94.83%	96.09%	97.52%	96.80%	15.98%
D-VAE-WGAN (C-sub)	96.29%	97.27%	98.12%	97.69%	11.11%
D-VAE-WGAN (C)	**97.89%**	**98.73%**	**98.64%**	**98.68%**	**5.11%**

由表9可以看出，D-VAE-WGAN(C)的检测性能明显高于其他检测模型；虽然D-VAE-WGAN(NC)中每个代理也使用了完整的训练集A进行训练，但由于没有采用交换参数的协同训练方式，致使模型的参数过于依赖训练数据自身所包含的信息，如样本输入的顺序及特征值变化趋势等，导致使用同样数据训练出的D-VAE-WGAN(NC)鲁棒性不足。另外，D-VAE-WGAN(C-sub)和D-VAE-WGAN (NC-sub)仅使用$\frac{1}{5}$的训练集训练，检测性能显著低于使用全训练集训练出的模型。然而通过观察发现，D-VAE-WGAN(C-sub)检测性能仅比使用全训练集的非协同模型D-VAE-WGAN(NC)略低一点，这是由于在训练时D-VAE-WGAN(C-sub)采用了交换参数的方式，使模型在少量样本的支撑下学习更多的数据分布，故可以表现出与全训练集训练出的非协同模型相近的检测效果。可见，协同训练的方法可以共享训练成果，使子网中的模型学习到更多数据分布，对于提升异常流量检测模型的性能有明显的促进作用。

4 结束语

本文提出一种基于深度学习的大规模SDN环境下异常流量的分布式检测方法。D-VAE-WGAN模型是由在云端服务器上的判别器和多个分布在控制器上的生成器构成的“一对多”分布式生成对抗网络架构。VAE作为编码器解决GAN收敛缓慢问题，使用Wasserstein距离作为相似性度量避免GAN“模式崩溃”风险。使用生成器交换参数的方式进行联合训练，使每个子网都能学习其他子网中的流量数据分布，提升SDN整体防御能力。在子网中构造检测代理，实现了对子网内部异常流量的分布式检测；避免了各控制器与云端服务器进行大规模数据交互，减少了共享链路的带宽消耗和服务器的计算压力。实验结果表明，所提方法可以快速准确地检测出InSDN等公开数据集中的异常样本，并且表现出比传统检测方法更高的检测精度；另外，由于仅使用正常流量训练，D-VAE-WGAN模型也具备对未知异常的检测能力。

未来计划将研究工作扩展到动态SDN中，如物联网、车联网等真实网络，并在研究中添加新兴的攻击类型和复杂攻击样本，以提升检测方法的泛化能力和适应性。

参考文献：

[1] ALHIJAWI B, ALMAJALI S, ELGALA H, et al. A survey on DoS/DDoS mitigation techniques in SDNs: classification, comparison, solutions, testing tools and datasets[J]. Computers and Electrical Engineering, 2022, 99: 107706.

[2] CHAHAL J K, BHANDARI A, BEHAL S. DDoS attacks & defense mechanisms in SDN-enabled cloud: taxonomy, review and research challenges[J]. Computer Science Review, 2024, 53: 100644.

[3] VERGARA J, GARZÓN C, BOTERO J F. A hybrid strategy for DoS attacks detection and Mitigation on SDN enabled real scenarioss[C]//Proceedings of the International Congress on Information and Communication Technology. Berlin: Springer, 2023: 705-714.

[4] BHAYO J, SHAH S A, HAMEED S, et al. Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks[J]. Engineering Applications of Artificial Intelligence, 2023, 123: 106432.

[5] 段雪源, 付钰, 王坤, 等. 基于简单统计特征的LDoS攻击检测方法[J].

通信学报, 2022, 43(11): 53-64.
DUAN X Y, FU Y, WANG K, et al. LDoS attack detection method based on simple statistical features[J]. Journal on Communications, 2022, 43(11): 53-64.

[6] JAFARIAN T, MASDARI M, GHAFFARI A, et al. A survey and classification of the security anomaly detection mechanisms in software defined networks[J]. Cluster Computing, 2021, 24(2): 1235-1253.

[7] 贾锟, 王君楠, 刘峰. SDN环境下的DDoS检测与缓解机制[J]. 信息安全学报, 2021, 6(1): 17-31.
JIA K, WANG J N, LIU F. DDoS detection and mitigation framework in SDN[J]. Journal of Cyber Security, 2021, 6(1): 17-31.

[8] VAN N D, HUY L D, TRUONG C Q, et al. Applying dynamic threshold in SDN to detect DDoS attacks[C]//Proceedings of the 2022 International Conference on Advanced Technologies for Communications (ATC). Piscataway: IEEE Press, 2022: 344-349.

[9] JASIM M N, GAATA M T. K-Means clustering-based semi-supervised for DDoS attacks classification[J]. Bulletin of Electrical Engineering and Informatics, 2022, 11(6): 3570-3576.

[10] CHENG Q M, WU C M, ZHOU H F, et al. Machine learning based malicious payload identification in software-defined networking[J]. Journal of Network and Computer Applications, 2021, 192: 103186.

[11] KUMAR R, AGRAWAL N. Software defined networks (SDNs) for environmental surveillance: a survey[J]. Multimedia Tools and Applications, 2024, 83(4): 11323-11365.

[12] KINGMA D P, WELLING M. Auto-encoding variational bayes[J]. arXiv Preprint, arXiv: 1312.6114v11, 2013.

[13] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial network[J]. arXiv Preprint, arXiv: 1406.2661v1, 2014.

[14] BAVANI K, RAMKUMAR M P, SELVAN G S R E. Statistical approach based detection of distributed denial of service attack in a software defined network[C]//Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). Piscataway: IEEE Press, 2020: 380-385.

[15] 周启钊, 于俊清, 李冬. SDN控制层泛洪防御机制研究: 检测与缓解[J]. 通信学报, 2021, 42(11): 41-53.
ZHOU Q Z, YU J Q, LI D. Research on flood defense mechanism of SDN control layer: detection and mitigation[J]. Journal on Communications, 2021, 42(11): 41-53.

[16] ZOLOTUKHIN M, KUMAR S, HÄMÄLÄINEN T. Reinforcement learning for attack mitigation in SDN-enabled networks[C]// Proceedings of the 2020 6th IEEE Conference on Network Softwarization (NetSoft). Piscataway: IEEE Press, 2020: 282-286.

[17] TAYFOUR O E, MARSONO M N. Collaborative detection and mitigation of DDoS in software-defined networks[J]. The Journal of Supercomputing, 2021, 77(11): 13166-13190.

[18] SATHEESH N, RATHNAMMA M V, RAJESHKUMAR G, et al. Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network[J]. Microprocessors and Microsystems, 2020, 79: 103285.

[19] SEBBAR A, ZKIK K, BADDI Y, et al. MitM detection and defense mechanism CBNA-RF based on machine learning for large-scale SDN context[J]. Journal of Ambient Intelligence and Humanized Computing, 2020, 11(12): 5875-5894.

[20] WANG K, FU Y, DUAN X Y, et al. Detection and mitigation of DDoS attacks based on multi-dimensional characteristics in SDN[J]. Scientific Reports, 2024, 14(1): 16421.

[21] SRI V G, NAGARAJAN R. A novel bidirectional LSTM model for network intrusion detection in SDN-IoT network[J]. Computing, 2024, 106(8): 2613-2642.

[22] YASER A L, MOUSA H M, HUSSEIN M. Improved DDoS detection utilizing deep neural networks and feedforward neural networks as autoencoder[J]. Future Internet, 2022, 14(8): 240.

[23] NOVAES M P, CARVALHO L F, LLORET J, et al. Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments[J]. Future Generation Computer Systems, 2021, 125: 156-167.

[24] WANG P, WANG Z X, YE F, et al. ByteSGAN: a semi-supervised generative adversarial network for encrypted traffic classification in SDN Edge Gateway[J]. Computer Networks, 2021, 200: 108535.

[25] SAMAAN S S, JEIAD H A. Feature-based real-time distributed denial of service detection in SDN using machine learning and Spark[J]. Bulletin of Electrical Engineering and Informatics, 2023, 12(4): 2302-2312.

[26] PATIL N V, KRISHNA C R, KUMAR K, et al. E-Had: a distributed and collaborative detection framework for early detection of DDoS attacks[J]. Journal of King Saud University - Computer and Information Sciences, 2022, 34(4): 1373-1387.

[27] SHUKLA P, KRISHNA C R, PATIL N V. SDDA-IoT: storm-based distributed detection approach for IoT network traffic-based DDoS attacks[J]. Cluster Computing, 2024, 27(5): 6397-6424.

[28] KAUR A, KRISHNA C R, PATIL N V. K-DDoS-SDN: a distributed DDoS attacks detection approach for protecting SDN environment[J]. Concurrency and computation: practice and experience, 2024, 36(3): 1-19.

[29] EZEH D A, DE OLIVEIRA J. An SDN controller-based framework for anomaly detection using a GAN ensemble algorithm[J]. Infocommunications Journal, 2023, 15(2): 29-36.

[30] PARRA G D L T, RAD P, CHOO K K R, et al. Detecting Internet of things attacks using distributed deep learning[J]. Journal of Network and Computer Applications, 2020, 163: 102662.

[31] FENG H F, ZHANG W T, LIU Y, et al. Multi-domain collaborative two-level DDoS detection via hybrid deep learning[J]. Computer Networks, 2024, 242: 110251.

[32] 肖警续, 郭渊博, 常朝稳, 等. 基于SDN的物联网边缘节点间数据流零信任管理[J]. 通信学报, 2024, 45(7): 101-116.
XIAO J X, GUO Y B, CHANG C W, et al. Zero trust management of

data flow between IoT edge nodes based on SDN[J]. Journal on Communications, 2024, 45(7): 101-116.

[33] 陈何雄, 罗宇薇, 韦云凯, 等. 基于联邦学习的SDN异常流量协同检测技术[J]. 计算机工程, 2023, 49(3): 168-176.
CHEN H X, LUO Y W, WEI Y K, et al. Collaborative detection technology of SDN abnormal traffic based on federated learning[J]. Computer Engineering, 2023, 49(3): 168-176.

[34] SHU J G, ZHOU L, ZHANG W Z, et al. Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(7): 4519-4530.

[35] 段雪源, 付钰, 王坤. 基于VAE-WGAN的多维时间序列异常检测方法[J]. 通信学报, 2022, 43(3): 1-13.
DUAN X Y, FU Y, WANG K. Multi-dimensional time series anomaly detection method based on VAE-WGAN[J]. Journal on Communications, 2022, 43(3): 1-13.

[36] ELSAYED M S, LE-KHAC N A, JURCUT A D. InSDN: a novel SDN intrusion dataset[J]. IEEE Access, 2020, 8: 165263-165284.

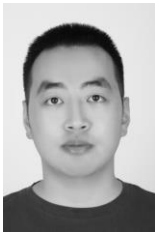
[37] KRISHNAN P, DUTTAGUPTA S, ACHUTHAN K. VARMAN: Multi-plane security framework for software defined networks[J]. Computer Communications, 2019, 148: 215-239.

[作者简介]

王坤（1981–），女，河南信阳人，海军工程大学博士生，信阳职业技术学院副教授，主要研究方向为网络安全、人工智能、信息对抗。

付钰（1982–），女，湖北武汉人，博士，海军工程大学教授、博士生导师，主要研究方向为信息安全、人工智能。

段雪源（1981–），男，河南开封人，博士，信阳师范大学讲师，主要研究方向为人工智能、信息处理、网络安全。



俞艺涵（1992–），男，浙江金华人，博士，海军工程大学讲师，主要研究方向为网络安全、运筹分析。

刘涛涛（1996–），男，江西吉水人，海军工程大学博士生，主要研究方向为网络安全、网络信息对抗。