文章编号：2096-1618(2025)03-0264-09

基于迁移学习的软件定义网络异常检测模型

肖德轩[1]， 秦 智[1,2]， 黄源源[1,2]， 卢嘉中[1,2]

(1. 成都信息工程大学网络空间安全学院，四川 成都 610225；2. 先进密码技术与系统安全四川省重点实验室，四川 成都 610225)

摘要：随着网络架构的不断演进，SDN已成为推动网络管理简化与通信创新的重要架构之一。然而，伴随着SDN在各个领域的广泛部署以及其结构日益复杂化，其在应对网络安全风险方面也面临着诸多挑战。大规模网络环境中的多样化攻击和海量数据制约了传统机器学习方法在该领域的进一步应用。而深度学习方法虽然在大规模数据处理方面具有优势，但其通常需要大量标记数据进行训练。因此提出一种异常检测模型，将改进的一维CBAM注意力机制与卷积神经网络相融合，以降低通道间的冗余并提高模型性能。同时，通过引入迁移学习方法，模型能够在仅使用有限标记数据训练的情况下有效识别SDN网络中的异常流量。实验结果显示，该模型在CIC-IDS2017数据集上取得了99.70%的准确率。在仅使用10%的SDN数据集中的标记数据进行微调的预训练模型达到98.53%的精确度，接近使用数据集80%进行训练的模型检测性能。这些结果验证了基于迁移学习和CNN的软件定义网络异常检测模型的可行性。

关 键 词：软件定义网络；卷积神经网络；深度学习；迁移学习；异常检测

中图分类号：TP309 **文献标志码**：A

doi：10.16836/j.cnki.jcuit.2025.03.002

0 引言

随着网络架构不断地发展与演进，软件定义网络(software defined network，SDN)已逐渐成为网络管理简化和推动通信与计算机网络创新的重要架构之一[1-3]。SDN的基本架构主要由控制平面和数据平面组成。控制平面集中了逻辑控制的功能，负责向数据平面发送指令以实现网络管理和控制。数据平面中的网络设备接收控制平面下发的转发规则并根据规则转发接收到的数据包[4]。然而，随着SDN在各个领域的广泛部署以及其结构日益复杂化，其面临的安全风险已成为一个不可忽视的重要问题[5]。

近年来，基于网络的异常检测系统(anomaly detection system，ADS)研究受到越来越多的关注[6]。目前，多种机器学习方法已被应用于构建基于异常的入侵检测系统的相关工作中。然而，随着网络攻击类别的多样化和网络流量的激增，传统机器学习等浅层学习技术已不再满足大规模基于网络的异常检测系统的要求[7]。深度学习被广泛用于处理不同环境中的各种入侵和安全问题，提高了网络空间的安全性[8]。

尽管先前的工作在开发异常检测系统方面取得了一些成功，但这些深度学习模型需要大量的标记数据进行训练，当训练数据不足时会对模型的准确率造成影响[9]。由于SDN主要部署在运营商主干网络和数据中心等场景中，网络数据量庞大且难以获取，因此，在SDN环境下获取大量的标记训练样本变得非常的困难。为解决标记样本不足的问题，提出一种基于迁移学习的SDN异常检测模型，能够在使用足够少量标记样本训练模型的情况下，仍然准确地识别SDN网络中的异常流量。

本文提出一种结合1D-ResNet和1D-DenseNet的一维卷积神经网络(convolutional neural networks，CNN)模型，可以直接从网络流量数据集的各项数据中提取特征而无须进行维度或特征变换，节省性能开销。在卷积神经网络模型中融合改进的一维CBAM注意力机制，改进的CBAM注意力模块通过改变通道注意力与空间注意力模块排布方式和结合MLP projector以减少通道冗余度。与CBAM注意力模块相比，结合改进的CBAM模块的模型表现出更好的性能。在SDN异常检测框架中结合迁移学习的思想，通过使用少量的SDN环境下标记样本进行训练，成功实现接近使用正常数据量进行训练的模型的检测性能。

1 相关工作

1.1 深度学习

近年来，许多研究团队分别提出基于深度学习的异常检测或入侵检测方法。Zhang等[10]提出一种基于高斯混合模型的欠采样聚类技术和合成少数过采样技术的SGM方法，用于处理大型数据集中的不平衡类

收稿日期：2023-11-14
基金项目：国家重点研发计划重点专项资助项目(2022YFB3103103)
通信作者：秦智. E-mail：mercyqz@cuit.edu.cn

别,但该模型对数据集中的 Web Attack Brute Force、Analysis 和 Backdoors 类别的样本检测率较低。Riyaz 等[11]提出一种被称为条件随机场和基于线性相关系数的特征选择算法,以选择贡献最大的特征并使用现有的卷积神经网络对其进行分类,但使用发布于 2000 年之前的 KDDCup-99 数据集评估其模型,旧数据集训练异常检测模型可能会出现问题。Andresini 等[12]提出一种通过执行最近邻搜索和聚类过程的组合方法来将网络流表示为二维图像,并使用生成的图像表示来训练深度学习模型,但产生了额外的性能开销,增加方法的复杂性。ElSayed 等[13]将一种结合卷积神经网络的权重矩阵的标准偏差的新正则化方法用于所提出的软件定义网络异常检测模型中,并在自制的 InSDN[14]数据集上进行实验和评估,然而并未在实验中将所提出的模型与广泛使用的机器学习或深度学习模型在公共数据集上进行比较。

综上所述,已有研究的共同特点是它们主要依赖传统网络环境下的大量标记数据来训练模型。在 SDN 网络环境下缺乏足够的标记数据时,模型的检测准确率会下降。

1.2　迁移学习

一些文献提出通过结合迁移学习的方法来减少样本不足对模型性能造成的影响。Sun 等[15]提出一种利用归纳迁移学习的流量分类方法,采用大量源域数据集和少量目标域标注数据集通过 TrAdaBoost 实现迁移,该方案基于机器学习的迁移方法,依赖于手动提取的特征进行分类。Guan 等[16]针对标记数据稀缺、计算能力有限的 5G 物联网场景提出一种基于深度迁移学习的流量分类方法,将流量数据转换为 IDX 图片文件格式对模型进行训练和测试,但 5G 网络和传统网络之间的数据分布可能存在显著差异(流量速率、连接数和数据包大小等),使用传统网络数据集代替真实的 5G 数据集可能无法充分反映 5G 网络中的复杂和多样的安全场景。Rodríguez 等[17]提出一种基于迁移学习、知识迁移和模型细化的针对数据集不平衡且稀缺的 5G 物联网场景的入侵检测框架,用于检测网络攻击,该方法缺乏与相关工作或模型进行的对比实验,无法充分说明该模型的性能表现。

综上所述,受现有研究的启发,本文设计了一种结合改进的一维 CBAM 注意力机制和卷积神经网络的异常检测模型,旨在提高对网络威胁和攻击的检测准确率,并借助迁移学习方法来解决在 SDN 环境中由于缺少大量标记数据而难以提升模型训练的质量问题。

2　异常检测方案

2.1　系统检测架构

迁移学习是机器学习领域中的一个重要分支,其核心思想是通过寻找源域和目标域之间的相似性,将源域中的一些能力和知识迁移到目标域,通过这种相似性的迁移帮助模型进行训练。迁移学习的数学定义如下:给定一个源域 $D_s=\{(X_s,Y_s)\}$ 和学习任务 T_s,其中 X_s 和 Y_s 分别表示源域的样本和标签。目标域 $D_t=\{(X_t,Y_t)\}$ 和学习任务 T_t,X_t 和 Y_t 分别表示目标域的样本和标签。迁移学习的目的是获取源域 D_s 和学习任务 T_s 中的知识以帮助提升目标域中的预测函数 f_t 的学习,其中 $D_s\neq D_t$ 或者 $T_s\neq T_t$。预测函数 f_t 定义为 $f_t(X_t)=Y_t$。T_t 和 Y_t 分别代表要解决的问题和对应域下的标记数据集[16-18]。

图 1 是结合迁移学习的 SDN 异常检测系统架构图。该系统主要包括 3 个部分:模型预训练、领域迁移和模型微调。

(1)模型预训练:需要在具有相对较多标记过的网络流量数据集和一般网络环境的源域(D_s)下,构建一种 CNN 和改进的一维 CBAM 注意力相融合的深度学习模型,模型的预训练和评估过程所需的数据集首先需要进行预处理。本文采用 SMOTE 过采样方法、NearMiss 欠采样方法、Z-score 数据标准化等数据预处理方法应对数据集中的类不平衡问题,再将预处理过的数据集应用于文中提出的深度学习模型中,最后在该模型上进行预训练和评估。

(2)领域迁移:模型利用在上一部分得到的一些知识,利用源域(D_s)和目标域(D_t)之间的相似性,选择具有最佳预测性能的模型和模型的权重参数作为迁移学习对象。迁移到缺乏丰富标记数据的 SDN 网络环境的目标域,将迁移学习模型作为目标域的 SDN 异常检测的预训练模型。

(3)模型微调:在目标域进行。在模型微调阶段,保留并冻结来自源域预训练模型的特征提取器的权重参数,并使用少量同样经过第一部分预处理方法处理的,在目标域下的 SDN 网络数据集来更新和微调深度学习模型的分类器部分,从而提高深度学习模型在目标域标记样本不足的情况下的检测性能。

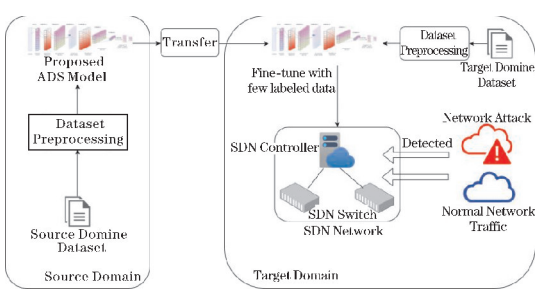


图 1　SDN 异常检测系统架构

2.2　模型结构

异常检测框架中提出的融合卷积神经网络和注意力机制的深度学习模型的整体结构如图 2 所示,其中卷积

神经网络采用一维 ResNet 网络[19]和 DenseNet 网络[20]。

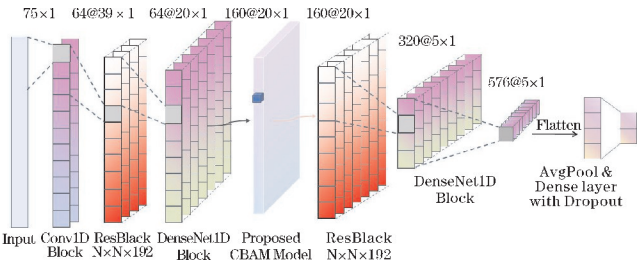


图 2 模型结构

模型的输入为网络流量数据集的 76 个一维特征，第一层包含 1 个一维卷积层和 1 个一维 MaxPool 层，第二层到第六层为交错排列的一维 ResNet 模块，DenseNet 模块和一个注意力模块，注意力模块的详细结构和参数将在下文中描述。第二层包括的 4 个输入和输出均为 64 个通道的 ResNet 块，第三层包含 8 个输入，输出维度均为 20 的 Dense Layer 和 1 个 Transition 层，第四层是一个注意力模块。以上 4 层模块共同构成模型的特征提取器部分，这 4 层的参数在模型迁移阶段被冻结，在迁移后的模型微调过程中参数不会更新。在注意力模块也有一个 ResNet 块和 DenseNet 块，所不同的是第二个 ResNet 块会使输出的通道数加倍，由 160 变为 320，并且输入的一维向量的长度由 20 缩短为 5，第二个 DenseNet 块则和第一个 DenseNet 块结构类似。模型的最后为输出层，包含 1 个 Flatten 层，2 个 Linear 层和 1 个参数为 0.5的 Dropout 层，用于输出模型分类预测的结果。在注意力模块之后的所有模块都会在模型迁移的微调阶段更新参数。

2.3 注意力模块

本文提出的注意力模型以 CBAM 注意力机制模块[21]为基础，在此之上改变注意力模块的输入和输出维度以匹配模型的一维输入，重新设计注意力模块的排布方式，并在之后加入 MLP projector 模块，这些改进可以增强模型的表征能力。此外，在模型中引入注意力机制有助于模型提高在小样本条件下分类的准确性[22]。

设中间特征图输入为 $\mathrm{Fin}\in R^{W\times C}$，作为输入，$C$ 为通道数量，W 为输入特征图的长度，则改进的 1D-CBAM 的整个过程可以写成

$$F_{\mathrm{output}}=F\oplus M_{\mathrm{MlpProjector}}((M_c(F)\otimes F)\oslash(M_s(F)\otimes F)) \tag{1}$$

其中$\oslash$表示元素在通道维度上拼接，$\oplus$表示元素之间的和，$\otimes$表示元素之间的乘积，F_{output}表示最终的输出。

(1)通道注意力模块：CBAM 模块的通道注意力模块首先通过同时执行最大池化和平均池化来提取通道特征，AvgPool1d 和 MaxPool1d 分别表示获取输入 F 的平均池化特征和最大池化特征。然后，将生成的两个特征传递到通道注意力模块的共享网络层。共享网络层由包含一个隐藏的激活层的多层感知机(multilayer perceptron，MLP)组成。之后，通过逐元素求和来合并输出特征向量，输出的特征向量经过 Sigmoid 激活函数后与原始输出相乘得到通道注意力模块的输出。

$$M_c(F)=\mathrm{Sigmoid}(W_{\mathrm{mlp}}(\mathrm{AvgPool1d}(F))+W_{\mathrm{mlp}}(\mathrm{MaxPool1d}(F))) \tag{2}$$

(2)空间注意力模块：1D-CBAM 注意力机制的空间注意力模块则是首先沿着通道维度进行最大池化和平均池化操作，得到 $S_{\mathrm{avg}}\in R^{w\times 1}$，$S_{\max}\in R^{w\times 1}$。然后使用卷积核大小为 7 的一维卷积层进行卷积，得到一维空间注意力特征图。空间注意力为

$$\begin{aligned}M_s(F)&=\mathrm{Sigmoid}(\mathrm{Conv1d}([\mathrm{Avgpool1d}(F);\mathrm{MaxPool1d}(F)]))\\&=\mathrm{Sigmoid}(W^7_{\mathrm{Conv1d}}([F^s_{\mathrm{Avg1d}};F^s_{\mathrm{Max1d}}]))\end{aligned} \tag{3}$$

最后，改进的一维 CBAM 注意力模块与原本的 CBAM 模块不同的是，它以并列的方式组合通道注意力模块和空间注意力模块的输出，并且在拼接通道注意力模块和空间注意力模块的输出后连接一个 MLP projector，得到改进的 CBAM 注意力模块的输出，改进的 CBAM 注意力模块如图 3 所示。MLP projector 的公式为

$$F(x)=fc_2(\mathrm{ReLU}(\mathrm{BatchNormal1d}(fc_1(x)))) \tag{4}$$

其中，fc_1 和 fc_2 代表两个全连接层。这种排布方式可以减少通道冗余，从而增强模型的特征表达能力和迁移能力[23]。本文参考相关研究[24]，在数学上通过使用通道间的皮尔逊相关系数对通道间的通道冗余进行评估，皮尔逊相关系数的计算公式如下：

$$\text{Pearson Correlation}=\frac{\sum_{i=1}^{d}\sum_{j=1}^{d}|\rho(i,j)|}{d^2} \tag{5}$$

$$\rho(i,j)=\frac{\sum_{n=1}^{N}(f_{n,i}-\bar{f}_i)(f_{n,j}-\bar{f}_j)}{\sqrt{\sum_{i=1}^{N}\|f_{n,i}-\bar{f}_i\|}\sqrt{\sum_{j=1}^{N}\|f_{n,j}-\bar{f}_j\|}} \tag{6}$$

其中，d 代表特征维度，$\rho(i,j)$代表特征通道 i 和 j 的皮尔逊相关系数。

如表 1 所示，在两个不同的 epoch 中，通过引入 MLP projector 并改变排列的方式后的 CBAM 模块的通道间皮尔逊相关系数相对更小。这一结果表明，改进后的 CBAM 模块能够减少通道之间的特征冗余，从而使模型具有更出色的性能。

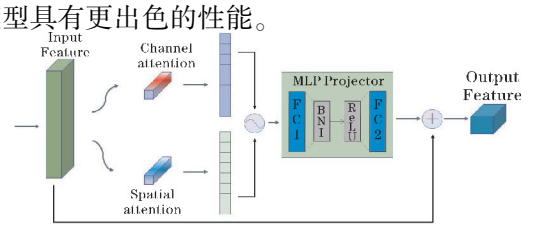


图 3 改进的 CBAM 注意力模块

表 1　预训练过程中通道冗余度的比较

Method	Epoch	皮尔逊相关系数
CBAM 模块	50	0.2216
经过改进的 CBAM 模块	50	0.2082
CBAM 模块	100	0.2170
经过改进的 CBAM 模块	100	0.1989

3　实验和结果分析

3.1　实验设置

实验主要分为两个部分:第一部分旨在评估所提出模型在 CISIDS2017 数据集上的性能,并与现有模型进行对比;第二部分采用迁移学习方法,在 InSDN 数据集上评估经过 CICIDS2017 数据集预训练后的模型借助少量目标域标记样本微调后的迁移效果,并与在该数据集上重新训练一个模型进行对比。实验中,为测试不同的标记数据量对模型微调效果的影响,分别选择数据集的0.5%、1%、5%、10% 和 80% 作为对比;实验均采用 Pytorch 框架,在模型的训练和预训练阶段,学习率设定为 3e-4,epoch 设置为 100。Batch size 在 CICIDS2017 数据集上进行评估和预训练时设置为 256,在迁移和微调阶段设置为 128。

3.2　数据集与数据预处理

3.2.1　数据集描述

CICIDS2017 数据集[25]由加拿大网络安全研究所收集和提出,包含 2813786 个网络流量样本,其中包括 14 种网络流量样本,每个样本包含 84 个特征维度。所有类别中,良性流量占 80.30%,攻击流量占 19.70%。样本涵盖了常见的攻击类型,如 DoS、DDoS、PortScan、Web 攻击等。CICIDS2017 数据集与其他公开数据集相比,更符合性能评估的 11 项标准[26],且该数据集是当前外部网络最具代表性的数据集,比其他数据集包含更多的特征、实例和网络攻击类型[27-28]。每个类别的数据分布如表 2 所示。

表 2　CICIDS2017 数据集样本比例

平衡前类别	样本数/个	百分比/%	平衡后类别	样本数/个	百分比/%
BENIGN	2260360	80.33	BENIGN	50000	21.52
DoS Hulk	229198	8.14	DoS Hulk	50000	21.52
PortScan	157703	5.60	PortScan	50000	21.52
DDoS	127082	4.52	DDoS	40,000	17.22
DoS GoldenEye	10289	0.36	DoS GoldenEye	10289	4.43
FTP-Patator	7894	0.28	FTP-Patator	7894	3.40
SSH-Patator	5861	0.21	SSH-Patator	5861	2.52
DoS slowloris	5771	0.20	DoS slowloris	5771	2.48
DoS Slowhttptest	5485	0.19	DoS Slowhttptest	5485	2.36
Bot	1943	0.07	Bot	3000	1.29
Infiltration	34	0.0012	Infiltration	500	0.22
Web-Attack Brute Force	1497	0.053			
Web-Attack XSS	648	0.023	Web-Attack	3500	1.51
Web-Attack Sql Injection	21	0.0007			

InSDN 数据集[14]则是一个用于在 SDN 环境下检测异常和攻击行为的数据集。InSDN 是首批收集软件定义网络数据并用于训练和验证基于 SDN 的入侵检测系统的数据集。该数据集包含 343889 条数据,涵盖 7 种攻击类型,包括 DoS、DDoS、Web-Attack 等。每个类别的详细数据分布如表 3 所示。

表 3　InSDN 数据集样本比例

类别	平衡前样本数	百分比/%	平衡后样本数/条	百分比/%
DDos	121942	35.46	15000	23.26
Probe	98129	28.54	15000	23.26
Normal	68424	19.90	10000	23.26
DoS	53616	15.59	10000	15.50
BFA	1405	0.41	3000	4.65
Web-Attack	192	0.06	500	4.65
BOTNET	164	0.05	500	4.65
U2R	17	0.005	500	0.78

3.2.2　数据预处理

为提高模型的性能,在模型训练之前需要对数据集进行预处理。数据预处理主要包含以下 3 个步骤:

(1)数据清洗:首先,将 CICIDS2017 数据集中存在的缺失值(NaN)值填充为零。其次,删除 CICIDS2017 和 InSDN 数据集中 Flow ID、Src Port、Dst Port、Src IP、Dst IP、TimeStamp 和 Protocol 等多余和重复的特征,将特征维度减少到 76。

(2)数据平衡:在 CICDIS2017 和 InSDN 数据集中,最大样本数量与最小样本数量之比分别为 107636 和 7173,表明这两个数据集均为不平衡数据集。如果直接使用不平衡数据集对模型进行训练,可能会导致模型有偏差和检测率低[29]。因此,使用结合了 SMOTE 过采样方法[30]和 Near Miss 欠采样技术[31]的数据平衡方法对不平衡数据集各类别样本的占比进行平衡。

(3)数据归一化:Z-score 标准化可以将数据特征的范围转换成均值为 0,标准差为 1 的标准正态分布,从而将不同尺度的特征值映射到相同的数值范围。一些研究表明,深度学习模型通常在进行归一化后的数据集上表现得更好[28,32]。Z-score 归一化的公式可以表示为

$$z=\frac{x-\mu}{\sigma} \tag{7}$$

其中,x 代表原始数据,z 代表经过归一化的数据,μ 代表原始数据的均值,σ 代表原始数据的标准差。

3.3 验证指标

为验证所提出的模型对异常流量的识别性能,使用以下常见的性能评估指标来衡量所提出的模型的表现,包括准确率(Accuracy)、精确率(Precision)、召回率(Recall)以及 F1 分数(F1-Score),计算公式如下:

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{8}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{9}$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{10}$$

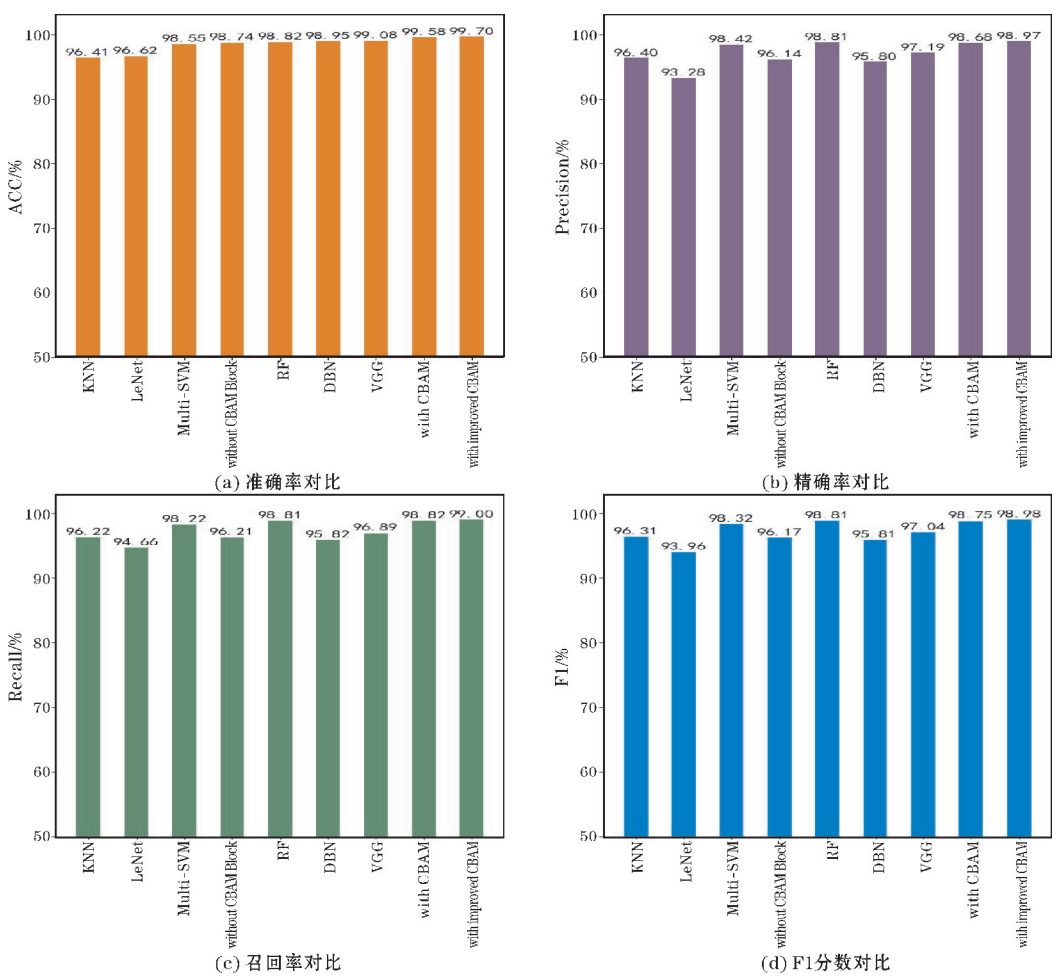
$$\text{F1}=\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{11}$$

其中,TP 代表被模型归类为正样本的正样本,TN 代表被模型归类为负样本的负样本,FP 代表被归类为正样本的负样本,FN 代表被归类为负样本的正样本。

3.4 模型在 CICIDS2017 数据集上的性能分析

为评估提出模型的性能,本文选择 6 种被广泛使用且具有明确的配置和良好效果的模型作为基线模型与所提出的模型进行对比。同时,将所提出的改进的 CBAM 模块与未改进的 CBAM 注意力模块在相同的配置下进行对比实验,以展示改进后的 CBAM 模块对模型性能的提升。

图 4 为本文所提出的深度学习模型和其他模型在 CICIDS2017 数据集上的图形视图对比,结果如表 4 所示。在 CICIDS2017 数据集上的多分类评估结果表明,本文提出的混合模型在所有测量中都比其他模型得出更高的结果,达到99.70% 的准确率和0.99的 F1 分数。所提出的模型在准确率指标下至少比相同配置下的其他验证方法高出0.62%,比使用未改进的 CBAM 模块的混合模型高出0.22%,在召回率和 F1 分数等方面相比其他模型也表现出更好的性能。实验结果表明,本文所提出的深度学习模型能够有效地检测网络中各种已知类型的攻击。



(a) 准确率对比 (b) 精确率对比

(c) 召回率对比 (d) F1分数对比

图 4 模型性能比较

表 4　模型在 CICIDS2017 数据集上的性能对比

模型	ACC/%	Precision/%	Recall/%	F1
KNN[25,28]	96.41	96.40	96.22	0.9631
LeNet[16,33]	96.62	93.28	94.66	0.9396
DBN[28,34]	98.95	95.80	95.82	0.9581
VGG16[35-36]	99.08	97.19	96.89	0.9704
Multi-SVM[28,37]	98.55	98.42	98.22	0.9832
RF[25,28]	98.82	98.81	98.81	0.9881
Our proposed model without CBAM Block	98.74	96.14	96.21	0.9617
Our proposed model with CBAM	99.58	98.68	98.82	0.9875
Our proposed model with improved CBAM	99.70	98.97	99.00	0.9898

本文提出的深度学习模型在 CICIDS2017 数据集上的分类结果如表 5 所示,模型的多分类混淆矩阵如图 5 所示。实验结果表明,Bot、Port Scan、DDoS、Dos Hulk 和 FTP-patator 等样本的检测率,F1 分数等指标较高,而其他类型的样本检测率相对较低。例如,Dos slowloris、Infiltration 和 Web-Attack,这些类别的 F1 分数均低于 98%,因为它们的数据分布与正常数据分布较为相似[28]。因此,尽管本文所提出的深度学习模型能以相对较高的准确率和 F1 分数检测大多数的攻击类型,但是如何提高与正常流量具有高度相似性的攻击样本的检测准确率仍然是一个值得研究的问题。

表 5　在 CICIDS2017 数据集上各类别的精确度、召回率和 F1

类别	Accuracy/%	Recall/%	Precision/%	F1
BENIGN	99.85	99.51	99.78	0.9964
Bot	100.0	99.83	100.0	0.9992
DDoS	99.94	99.79	99.85	0.9982
Dos GoldenEye	99.97	99.81	99.56	0.9968
Dos Hulk	99.97	99.93	99.91	0.9992
Dos slowloris	99.90	98.91	96.88	0.9788
Dos Slowhttptest	99.95	99.31	98.71	0.9901
FTP-patator	99.99	99.81	100.0	0.9990
Infiltration	99.98	94.00	96.91	0.9543
Port-Scan	99.99	99.96	99.98	0.9997
SSH-patator	99.95	99.15	98.89	0.9902
Web-Attack	99.93	97.71	97.57	0.9764

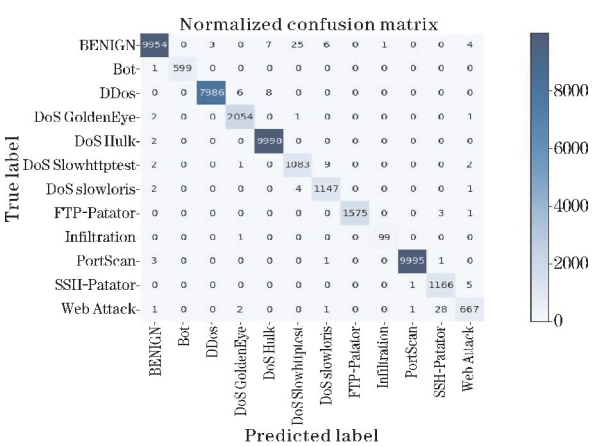


图 5　多分类混淆矩阵

3.5　迁移学习结果分析

图 6 为 2 个不同模型在迁移实验中得到的模型的精度折线图,图 7 为 2 个模型的召回率折线图。橙色折线代表在不同训练样本数量下经过迁移和微调得到的模型,紫色折线则表示在相同场景下,没经过预训练的新模型。当训练样本占整个数据集的0.5%时,经过预训练的迁移模型的平均精度为90.78%,平均召回率为86.76%,而新模型的平均准确率和召回率分别为80.87%和62.25%。当训练样本达到占整个数据集的10%时,经过预训练的迁移模型的平均精度为98.53%,平均召回率为95.82%,新模型的平均准确率和召回率分别为95.29%和94.06%。可以发现,当训练数据的百分比低于 5%时,经过预训练得到的模型精度显著优于没有经过预训练的模型,这意味着本文所提出的基于迁移学习方法的预训练模型在稀缺数据集场景下的表现更好。并且,经过预训练的迁移学习模型,在使用数据集中的 10%数据进行微调后,在测试集上的准确度仅比使用 80%数据集的未经预训练的新模型低 1%左右,远高于使用相同的 10%数据集的数据量进行训练的新模型。结果表明,结合迁移学习和微调的训练方法仅使用 10%的有限标记数据进行训练就能够实现与完整训练数据相近的结果。

此外,如图 8 所示,通过 2 个模型之间的训练时间对比可以发现,使用迁移学习和微调方法的模型的训练时间短于重新训练一个模型,并且随着标记样本的增加,训练时间的差距会更加明显。这表明尽管基于大规模数据集进行训练的模型在某些特定任务上可能取得更优表现,但是会付出更多的时间成本。在某些情况下,即便存在充足的标记数据,迁移学习依然是一个值得考虑的解决方案。

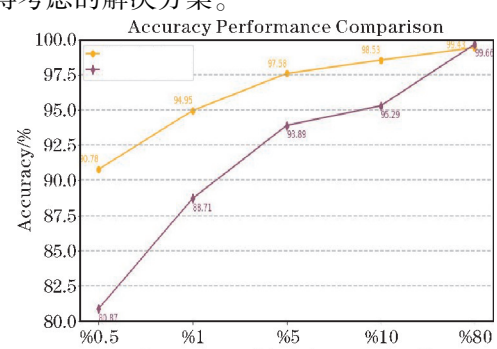


图 6　迁移学习模型与重新训练模型的准确性对比

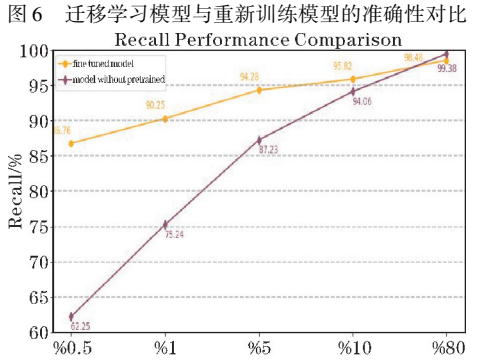


图 7　迁移学习模型与重新训练模型的召回率对比

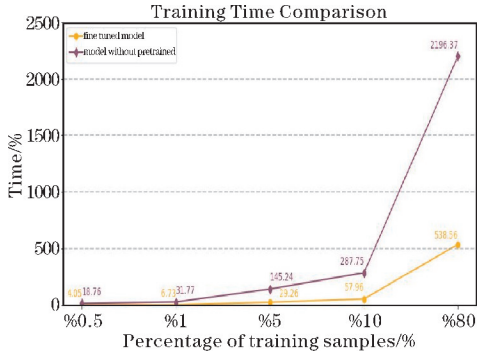


图 8 迁移学习模型与重新训练模型的训练时间对比

表 6 为基于迁移学习方法的模型在不同训练样本场景下的多分类实验结果,分别展示各个类别的准确率、召回率和 F1 分数。实验结果显示,尽管 U2R 和 Web-Attack 类别的准确率相对较高,但是召回率和 F1 分数这两项评估指标的得分普遍低于其他类别。这表示在缺乏训练样本时,U2R 和 Web-Attack 这 2 个类别受到的影响较为显著。这是由于这些攻击往往采用与正常 SDN 网络流量类似的传输协议和通信方式,借此绕过监测系统。因此,在缺乏相关样本的情况下,模型难以将这些流量与正常流量区别开来。

表 6 迁移学习模型在不同样本数量下各类别的精确率、召回率和 F1

占比	评价指标	BFA	BOTNET	DDoS	Dos	Normal	Probe	U2R	Web-Attack
	Accuracy/%	96.77	99.12	98.94	96.16	95.06	97.07	99.10	99.04
0.5%	Recall/%	62.65	4.62	97.53	92.05	88.48	98.24	2.01	13.08
	F1	0.6811	0.0883	0.9806	0.8980	0.8680	0.9486	0.0394	0.1997
	Accuracy/%	96.53	99.67	99.24	96.28	96.03	98.72	99.28	99.26
1%	Recall/%	66.94	75.96	97.55	95.48	88.33	98.71	21.01	19.60
	F1	0.6997	0.8077	0.9861	0.9040	0.8909	0.9769	0.3472	0.3266
	Accuracy/%	99.01	99.97	99.94	99.28	98.70	99.41	99.78	99.42
5%	Recall/%	92.95	99.58	99.83	98.22	96.96	99.40	78.95	38.53
	F1	0.9117	0.9854	0.9988	0.9805	0.9647	0.9893	0.8681	0.5504
	Accuracy/%	99.36	99.99	99.94	99.46	99.21	99.67	99.86	99.55
10%	Recall/%	91.81	99.33	99.82	98.41	99.02	99.56	0.8689	71.11
	F1	0.9403	0.9933	0.9989	0.9853	0.9787	0.9940	0.9168	0.7451
	Accuracy/%	99.36	100.00	100.00	99.79	99.77	99.82	99.98	99.50
80%	Recall/%	95.83	100.00	100.00	99.50	99.55	99.67	99.00	58.00
	F1	0.9426	1.00	1.00	0.9943	0.9938	0.9967	0.9900	0.6824

4 结束语

本文提出一种基于迁移学习的异常检测模型,用于检测 SDN 网络中不同类型的网络攻击,以应对不断增加的 SDN 安全威胁。该模型融合了卷积神经网络和改进的一维 CBAM 注意力机制,通过结合 MLP Projector 及重新设计原本的 CBAM 中通道注意力和空间注意力模块的排列方式,减少通道冗余,提高模型的性能。与被广泛使用且具有良好效果的基线模型相比,本文提出的模型在实验中取得了最佳的多分类性能。同时,为解决 SDN 领域标记数据不足的问题,本文结合迁移学习方法,通过对模型进行预训练和微调来实现源域和目标域之间的迁移。实验结果表明,仅使用 5% ~10% 的 SDN 数据集进行微调的预训练模型的检测效果接近使用 80% 数据集进行训练的新模型。在未来的工作中将计划进一步收集 SDN 网络环境中的攻击流量,并深入研究与正常流量具有高度相似性的攻击样本和异常检测模型。

参考文献:

[1] Farhady H, Lee H Y, Nakao A. Software-defined networking: A survey [J]. Computer Networks, 2015,81:79-95.

[2] Masoudi R, Ghaffari A. Software defined networks: A survey[J]. Journal of Network and computer Applications,2016,67:1-25.

[3] Ahmad I, Namal S, Ylianttila M, et al. Security in software defined networks: A survey [J]. IEEE Communications Surveys & Tutorials, 2015, 17(4):2317-2346.

[4] Scott-Hayward S, Natarajan S, Sezer S. A survey of security in software defined networks [J]. IEEE Communications Surveys & Tutorials, 2015, 18(1):623-654.

[5] Kalkan K, Gur G, Alagoz F. Defense mechanisms against DDoS attacks in SDN environment [J]. IEEE Communications Magazine, 2017, 55 (9): 175-179.

[6] Kumar S. Survey of current network intrusion detection techniques [J]. Washington Univ. in St. Louis, 2007: 1-18.

[7] Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: A survey [J]. applied sciences, 2019, 9(20): 4396.

[8] Lee S W, Mohammadi M, Rashidi S, et al. Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review[J]. Journal of Network and Computer Applications, 2021, 187: 103111.

[9] Jurcut A, Niculcea T, Ranaweera P, et al. Security considerations for Internet of Things: A survey[J]. SN Computer Science, 2020, 1: 1-19.

[10] Zhang Hongpo, Huang Lulu, Wu ChaseQ. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset [J]. Computer Networks, 177; 2020, (7): 107315.

[11] Riyaz B, Sannasi Ganapathy. A deep learning approach for effective intrusion detection in wireless networks using CNN [J]. Soft Computing 2020, 24: 17265-17278.

[12] Andresini Giuseppina, Annalisa Appice, Donato Malerba. Nearest cluster-based intrusion detection through convolutional neural networks[J]. Knowledge-Based Systems, 2021, 216: 106798.

[13] ElSayed Mahmoud Said. A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique [J]. Journal of Network and Computer Applications 2021, 191: 103160.

[14] Elsayed Mahmoud Said, Nhien-An Le-Khac, Anca D Jurcut. InSDN: A novel SDN intrusion dataset [J]. Ieee Access 2020, 8: 165263-165284.

[15] Sun G, Liang L, Chen T, et al. Network traffic classification based on transfer learning[J]. Computers & electrical engineering, 2018, 69: 920-927.

[16] Guan J, Cai J, Bai H, et al. Deep transfer learning-based network traffic classification for scarce dataset in 5G IoT systems[J]. International Journal of Machine Learning and Cybernetics, 2021, 12(11): 3351-3365.

[17] Rodríguez E, Valls P, Otero B, et al. Transfer-learning-based intrusion detection framework in IoT networks[J]. Sensors, 2022, 22(15): 5621.

[18] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks? [J]. Advances in neural information processing systems, 2014, 27.

[19] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[20] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks [C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[21] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]. Proceedings of the European conference on computer vision (ECCV), 2018: 3-19.

[22] Liu J, Zhang K, Wu S, et al. An investigation of a multidimensional CNN combined with an attention mechanism model to resolve small-sample problems in hyperspectral image classification[J]. Remote Sensing, 2022, 14(3): 785.

[23] Zbontar J, Jing L, Misra I, et al. Barlow twins: Self-supervised learning via redundancy reduction [C]. International Conference on Machine Learning. PMLR, 2021: 12310-12320.

[24] Wang Y, Tang S, Zhu F, et al. Revisiting the transferability of supervised pretraining: an mlp perspective [C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 9183-9193.

[25] Sharafaldin I, Lashkari A H, Ghorbani A A. Toward generating a new intrusion detection dataset and intrusion traffic characterization[J]. ICISSp, 2018, 1: 108-116.

[26] Gharib A, Sharafaldin I, Lashkari A H, et al. An evaluation framework for intrusion detection dataset[C]. 2016 International Conference on Information Science and Security (ICISS). IEEE, 2016: 1-6.

[27] Rosay A, Carlier F, Leroux P. Feed-forward neural

network for Network Intrusion Detection [C]. 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring). IEEE, 2020: 1-6.

[28] Yang L, Moubayed A, Shami A. MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles [J]. IEEE Internet of Things Journal, 2021, 9(1): 616-632.

[29] Chen Z, Yan Q, Han H, et al. Machine learning based mobile malware detection using highly imbalanced network traffic [J]. Information Sciences, 2018, 433: 346-364.

[30] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique [J]. Journal of artificial intelligence research, 2002, 16: 321-357.

[31] Mani I, Zhang I. kNN approach to unbalanced data distributions: a case study involving information extraction [C]. Proceedings of workshop on learning from imbalanced datasets. ICML, 2003, 126 (1): 1-7.

[32] Ali Alheeti K M, McDonald-Maier K. Intelligent intrusion detection in external communication systems for autonomous vehicles [J]. Systems Science & Control Engineering, 2018, 6(1): 48-56.

[33] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86 (11): 2278-2324.

[34] Elmasry W, Akbulut A, Zaim A H. Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic [J]. Computer Networks, 2020, 168: 107042.

[35] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv: 1409. 1556, 2014.

[36] Yang L, Shami A. A transfer learning and optimized CNN based intrusion detection system for Internet of Vehicles [C]. ICC 2022-IEEE International Conference on Communications. IEEE, 2022: 2774-2779.

[37] Vijayanand R, Devaraj D, Kannapiran B. Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection [J]. Computers & Security, 2018, 77: 304-314.

[38] Pan S J, Yang Q. A survey on transfer learning [J]. IEEE Transactions on knowledge and data engineering, 2009, 22(10): 1345-1359.

A Software Defined Network Anomaly Detection Model based on Transfer Learning

XIAO Dexuan[1], QIN Zhi[2], HUANG Yuanyuan[1,2], LU Jiazhong[1,2]

(1. College of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China; 2. Advanced Cryptography & System Security Key Laboratory of Sichuan Province, Chengdu 610225, China)

Abstract: With the continuous evolution of network architecture, SDN has become one of the important architectures to promote network management simplification and communication innovation. However, with the extensive deployment of software-defined networks in various fields and its increasingly complex structure, SDN faces many challenges in dealing with network security risks. The diversified attacks and massive data in large-scale network environments restrict the further application of traditional machine-learning methods in this field. Although the deep learning method has advantages in large-scale data processing, it usually needs a large number of labeled data for training. Therefore, this paper proposes an anomaly detection model, which combines the improved one-dimensional CBAM attention mechanism with a convolutional neural network to reduce the redundancy between channels and improve the performance of the model. At the same time, by introducing the transfer learning method, the model can effectively identify the abnormal traffic in the SDN network with only limited labeled data training. The experimental results show that the model achieves 99.70% accuracy on the cicids 2017 data set. The accuracy of the pre-training model using only 10% of the labeled data in the SDN dataset for fine-tuning is 98.53%, which is close to the detection performance of the model using 80% of the dataset for training. These results verify the feasibility of software-defined network anomaly detection model based on transfer learning and CNN.

Keywords: software defined network; convolutional neural network; deep learning; transfer learning; anomaly detection