

一种物联网环境下的分布式异常流量检测方案

丁庆丰,李晋国

(上海电力大学 计算机与科学技术学院,上海 201306)

摘要: 物联网终端设备数量的急剧增加带来了诸多安全隐患,如何高效地进行异常流量检测成为物联网安全研究中的一项重要任务。现有检测方法存在计算开销大的问题,且不能显式地捕捉流量数据中的关系和结构,难以应对新型网络攻击。考虑网络结构和节点设备之间的复杂通信模式,提出一种基于图神经网络的分布式异常流量检测方案。结合物联网环境对卷积神经网络进行改进,识别节点之间的复杂关系,同时在物联网设备、转发器和雾节点上设计并部署分布式检测单元,通过分布式检测架构实现本地化的异常流量检测,从而降低检测延迟和时间开销。在此基础上,引入注意力模块强化对关键特征的提取,增强模型的可解释性,进一步提高检测精度。在公开数据集CTU-13上的实验结果表明,该方案准确率和AUC值达到99.93%和0.99,只需9.26 s即可完成检测,且带宽消耗仅为845 kb/s。
关键词: 物联网;异常流量检测;图神经网络;注意力机制;多层感知机;分布式系统

开放科学(资源服务)标志码(OSID):



中文引用格式:丁庆丰,李晋国.一种物联网环境下的分布式异常流量检测方案[J].计算机工程,2022,48(8):152-159.

英文引用格式:DING Q F,LI J G.A distributed abnormal traffic detection scheme in Internet of Things environment[J].Computer Engineering,2022,48(8):152-159.

A Distributed Abnormal Traffic Detection Scheme in Internet of Things Environment

DING Qingfeng,LI Jinguo

(School of Computer Science and Technology,Shanghai University of Electric Power,Shanghai 201306,China)

[Abstract] The sharp increase in the number of Internet of Things (IoT) terminal devices has introduced many security risks. The effective detection of abnormal traffic has become an important task in the security research of IoT. Existing detection methods have high computational overhead and cannot explicitly capture the relationship or structure in the traffic data; thus, handling new network attacks is difficult. Considering the network structure and complex communication mode between node devices, a distributed abnormal traffic detection scheme based on Graph Neural Network (GNN) is proposed. Combined with an IoT environment, Convolutional Neural Network (CNN) is improved to identify the complex relationships between nodes. Simultaneously, a distributed detection unit is designed and deployed on the IoT devices, transponders, and fog nodes, and the localized abnormal traffic detection is realized through the distributed detection architecture, thereby reducing the detection delay and time overhead. On this basis, an attention module is introduced to strengthen the extraction of key features, enhance the interpretability of the model, and further improve the detection accuracy. Experiments on the CTU-13 public dataset show that the accuracy and AUC value of this scheme achieve 99.93% and 0.99, respectively. Detection can complete in only 9.26 s, and the bandwidth consumption is only 845 kb/s.

[Key words] Internet of Things (IoT); abnormal traffic detection; Graph Neural Network (GNN); attention mechanism; Multi-layer Perceptron (MLP); distributed system

DOI:10.19678/j.issn.1000-3428.0063284

0 概述

随着物联网生态系统的发展,大量的智能终端设备被广泛应用于多个物联网应用领域,如智能家

居、智慧医疗、智慧交通4.0等^[1]。然而,物联网终端设备数量的急剧增加带来了诸多严峻的安全隐患,复杂的网络环境使得物联网设备产生的数据容易遭到泄露、攻击或者中断。一方面,物联网终端设备往

基金项目:国家自然科学基金(U1936213,61702321)。

作者简介:丁庆丰(1996—),男,硕士研究生,主研方向为信息安全、网络入侵检测;李晋国,副教授、博士。

收稿日期:2021-11-19 修回日期:2022-01-16 E-mail:philding@foxmail.com

往受计算、内存、带宽等资源的限制,其自身的局限性给物联网带来了更高要求的安全挑战;另一方面,物联网设备之间存在着较为密切的关联性,一旦有设备遭到入侵,可能会导致用户隐私数据泄露、网络基础设施无法正常运行、网络拥堵或瘫痪等状况,甚至会造成巨大的经济损失和社会损失,严重威胁企业 and 国家安全^[2]。

在过去几年中,机器学习和深度学习的兴起与发展推动了物联网安全领域的研究,各种类型的神经网络(如卷积神经网络、长短期记忆网络、自动编码器)大量应用于物联网的入侵检测中^[3]。文献[4]提出一种基于3个不同分类器(决策树、JRP算法、随机森林)的分级入侵检测系统(Intrusion Detection System, IDS),前两个分类器并行运行并将结果馈送到第3个分类器中,在IDS2017数据集中取得了较好的效果,但是系统模型相对简单,准确率和误报率不理想。文献[5]为了解决不平衡样本的问题,基于卷积神经网络提出一种CNN-FDC方法,将KDD-CUP99数据集转化为灰度图像后,用焦距损失替换原损失函数,弱化了攻击样本较少带来的影响,但是该模型在面对高维数据时往往精度较低。文献[6]提出一种混合深度学习模型CNN-LSTM,利用长短期记忆网络来学习高维流量数据的时间特征。与其他先进的入侵检测算法相比,该模型虽然取得了99.03%的准确率,但主要缺点是使用基于反向传播随机梯度的方法来更新权重,这需要较长的时间进行训练和更新,无法满足物联网系统的低时延性要求,且运算开销大。文献[7]引入自然语言处理领域比较热门的Transformer方法,结合流量数据集对模型进行改进,提高了检测精度并降低了时延,但是训练的样本是基于统计学的攻击样本,当遇到传播式攻击时检测效果较差,无法满足物联网系统的高动态性要求。

机器学习和深度学习方法大多被应用在邻居节点固定的欧式空间上,然而在真实的物联网场景中,大量的边缘设备和传感器会以复杂的、非线性的方式连接在一起,从而构成相邻节点不固定的非欧式空间^[8],但多数传统方法是浅层学习方法,仅从统计学角度分析单个节点的流量数据的异常,并没有显式地学习变量之间现有的关系或者结构,因而常规的深度学习方法在处理非欧式空间数据上的表现仍难以使人满意。一些狡猾的入侵者会用低强度、高针对性的异常流量发起攻击,其中数据包与合法流量非常相似,在统计分析层面不会造成重大变化^[9],这类新型攻击往往很难被传统方法所检测到。

本文在图神经网络的基础上,提出一种适用于物联网环境的分布式异常流量检测方案。对图卷积神经网络(Graph Convolution Neural Network, GCN)进行改进,去除网络中的消息传递模块,在计算损失函数时隐式地引入邻接信息并结合多层感知机(Multi-Layer Perceptron, MLP)进行训练,同时设计2个Graph-MLP(下文简称GMLP)简化计算并方便部署分布式多级异常检测单元。分布式流量异常

检测体系结构将在每个物联网的活动节点(如雾节点或者SDN转发器)上进行异常检测,相邻的传感器之间相互交换信息,以协同的方式保护网络,其中边缘GMLP用于特征分类并预测相邻节点发生异常的概率,而节点GMLP用于节点特征更新并产生自身异常状态的概率。在此基础上,引入注意力机制计算每个相邻节点的关注度,通过优化每个全连接层的权重选择过程,进一步提升模型的检测精度。

1 基于GCN的分布式物联网异常流量检测

本文考虑在雾节点的一侧或是在SDN边缘转发器上部署一个AI分布式检测模块,来替代以往运行在虚拟服务器或者云上的检测方法。这些检测模块由边缘转发器上的低功耗AI处理器来实现。每个分布式单元关注数据传输业务的子集,包括检测邻居节点的模块信息和异常状态,最终实现本地化的异常流量检测。

1.1 分布式异常流量检测架构

如图1所示,描述一个通用的物联网架构体系来支持本文所提出的分布式物联网异常检测方案的实现,该基础设施可以适用于多个物联网应用领域。由图1可知,整个架构由物联网设备、雾节点和云计算3层物联网设施组成,具有云-边-端3层分布式体系^[10]。其中:云计算层位于整个架构的最高层,为整个网络提供高速海量的计算资源和分析能力;边缘计算层位于终端设备和云计算层中间,节点包括网关、路由器以及边缘服务器;终端节点处于网络最边缘,包括各类传感器和移动设备,负责收集和转发各类原始数据^[11]。在多数物联网场景下,如果将处理时间敏感数据的应用程序运行在雾节点上,将大量原始数据传输到中央云计算层处理,不仅会带来传输的延迟,而且会导致额外的计算开销^[12]。

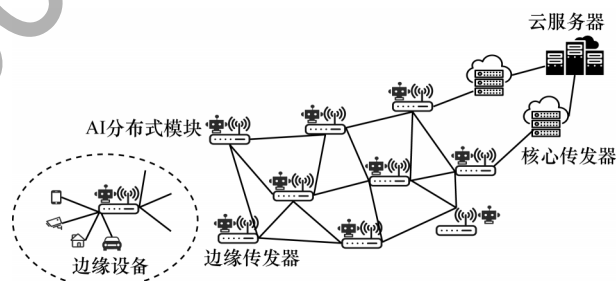


图1 物联网分布式异常流量检测架构

Fig.1 Distributed abnormal traffic detection architecture of IoT

为了减少资源开销并优化检测效果,可将物联网设备组成的网络看作是由节点设备之间通信通道组成的图,其结构和行为可以类比为图中各个节点的边,而网络中的异常行为可以看作是异常节点或者是异常边,则由图神经网络可以直接处理图结构的数据。利用这一特点,可以提取网络中出现的许多结构化特征,并且利用它们对节点进行分类。将每一个SDN转发器和核心转发器看作是图的节点,每个节点对应着一个描述节点行为的标签,而将它之间的底层连接看成边,从而可以将异常流量检测的任务看作是检测图中的异常节点或者异常边。

1.2 图卷积神经网络

图卷积神经网络可以学习静态网络中的内部关系,包括网络中的节点如何在正常行为模式中相互作用,以及如何与进出站的流量进行通信。在本文方案中,图卷积神经网络用于学习网络流之间的时空关系。给定图关系公式 $G=(V,E)$,其中 $E \in R[|V| \times |V|]$ 是所有顶点的集合,为邻接矩阵,其元素表示各个顶点之间的连接关系。

以图2(a)为代表的欧式空间卷积操作实际上是使用固定大小可学习的卷积核来抽取像素的特征。与这种方法不同,图结构中的邻居节点是不固定的,因此,研究人员将频谱的处理方式扩展到图(非欧几里得空间)上,如图2(b)所示。

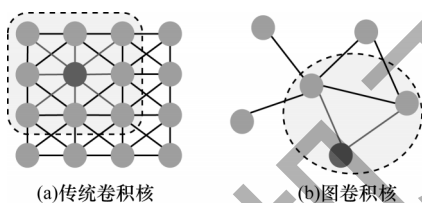


图2 传统卷积核与图卷积核

Fig.2 Traditional convolution kernel and graph convolution kernel

图上的频域卷积主要是利用图傅里叶变换来实现卷积,可以定义为输入信号 $x \in \mathbb{R}^N$ 与傅里叶域中过滤器 $g = \text{diag}(\theta)$ 的乘积,如式(1)所示:

$$g * x = U((U^T g) \odot (U^T x)) = U_{g\theta} \Lambda U^T x \quad (1)$$

在式(1)中, U 是由归一化拉普拉斯矩阵的特征向量组成的矩阵,计算公式如下:

$$L = I_N - D^{-\frac{1}{2}} X A = U \Lambda U^T \quad (2)$$

其中: I_N 是大小为 $N \times N$ 的矩阵; A 是邻接矩阵; 度矩阵元素定义为 $D_{ij} = \sum_i A_{ij}$ 。

为了降低计算成本,一些研究人员采用切比雪夫多项式 $T_k(x)$ 的 K 阶截断逼近 $g_{\theta}^{(k)}$ 的方法,从而简化了用切比雪夫多项式计算 $g * x$ 的方法,如式(3)所示:

$$g * x \approx \theta \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) \quad (3)$$

经过上述推导,最终的图卷积神经网络表达式为:

$$X^{(l+1)} = \sigma(\tilde{A} X^{(l)} W^{(l)}) \quad (4)$$

其中:第 l 层的特征是 $X^{(l+1)} \in \mathbb{R}^{N \times D}$, N 是节点数, D 是特征向量表示的维度; $W^{(l)}$ 是待训练的第 l 个单个线性层的权重; σ 是激活函数; 归一化矩阵 \tilde{A} 用来传递邻居节点之间的信息。

$$\tilde{A} = D^{-\frac{1}{2}} (A + I_N) D^{-\frac{1}{2}} \quad (5)$$

GCN模型的输入是一个图,其中包含 M 个具有特征的任意节点,到该节点的连接作为该图的边。使用GCN检测模型,目标是学习图上特征的函数来对输入节点进行分类^[13]。在常规的图卷积神经网络中,训练和分类过程都依赖于消息传递模块,即通过

邻接矩阵和特征相乘来显式地学习图的结构信息,多层感知机和连通节点上的求和运算被结合在一起在邻居之间传播特征^[14]。因此,图神经网络模型开销往往比较大,不利于部署在一些对性能和效率要求比较高的应用环境中。

1.3 改进的图多层感知机模型

如上文所述,以往的图模型重点关注如何利用邻居之间的消息传递,通过节点之间的信息交换,在前馈特征传播中显式地学习图的特征信息。但是这些频繁的消息传递往往会导致复杂的结构和复杂的计算,不适用于物联网环境中的网络异常流量检测。因此,本文引入一种新的思路来学习节点的特征变换,以避免显式的消息传递。整体结构上,以多层感知机为基础,将其与一种新的对比损失函数相结合来补偿在图数据建模中的显示结构,以监督节点特征变换的学习。

1.3.1 基于MLP的图模型结构

图3展示了本文改进的图多层MLP模型结构,其中最底层是特征处理模块,原始的节点特征 X 经过一组经典的神经网络处理模块(线性层-激活函数-层归一化-Dropout)后得到预处理的节点特征表示 \tilde{X} 。随后 \tilde{X} 被用于2个部分计算模型的损失函数: \tilde{X} 经过一层线性变化得到特征表示 Z 用于计算邻域对比损失; Z 经过一层线性变化得到特征表示 Y (此处为模型输出层,维度与分类数目相同)用于计算交叉熵损失(Cross Entropy, CE)损失。

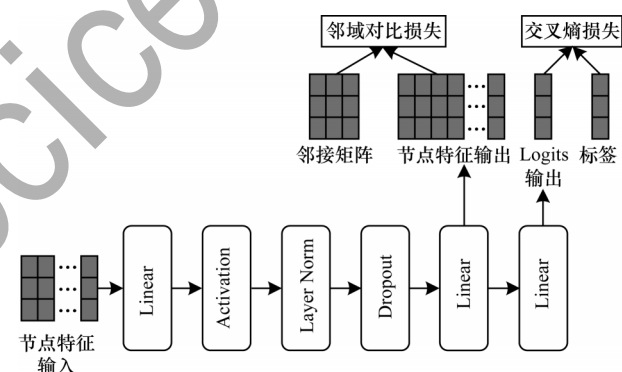


图3 图多层感知机模型结构

Fig.3 Structure of graph multi-layer perception model

整个模型可以表示为:

$$X^{(1)} = \text{Dropout}(\text{LN}(\sigma(XW^0))) \quad (6)$$

$$Z = X^{(1)} W^1 \quad (7)$$

$$Y = Z W^2 \quad (8)$$

1.3.2 监督节点特征转换的对比邻域损失

为了有效地提取图连接信息特征,模型中的损失函数应遵循同质性原则,即在特征表示空间中,连接的节点应是相似的,不连接的节点应是远离的,这和对比学习中的理念非常吻合。基于这样的动机,本文应用邻域对比损失函数,使得基于MLP的模型无需显式的消息传递模块即可学习图节点的连接特征。

在邻域对比损失函数中,对于每个节点,其 r 阶邻居被认为是正样本,其他节点被认为是负样本。该损失函数鼓励正样本更接近目标节点,并在特征距离方面将负样本推离目标节点。具体而言,第 i 个节点的邻域对比损失可以表示为:

$$\ell = -\lg \frac{\sum_{j=1}^B 1_{[j \neq i]} \gamma_{ij} \exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^B 1_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (9)$$

其中: B 表示batchsize; $\text{sim}()$ 是余弦相似度计算函数; τ 是常数; γ_{ij} 表示节点 i 和节点 j 之间的连接强度,将其计算为归一化邻接矩阵 A 的 r 次方 \tilde{A}_r ,当且仅当节点 j 是节点 i 的 r 阶邻居, γ_{ij} 才为非0值。

$$\begin{cases} \gamma_{ij} = 0, & \text{节点}j\text{不是节点}i\text{的}r\text{阶邻居} \\ \gamma_{ij} \neq 0, & \text{节点}j\text{是节点}i\text{的}r\text{阶邻居} \end{cases} \quad (10)$$

最终,模型的损失函数由Ncontrast损失和CE损失两部分组成:

$$l_{\text{NC}} = \alpha \frac{1}{B} \sum_{i=1}^B \ell_i \quad (11)$$

$$l_{\text{final}} = l_{\text{CE}} + l_{\text{NC}} \quad (12)$$

其中: l_{CE} 是交叉熵损失; α 是平衡这2个损失的加权系数。

1.3.3 模型训练

整个模型以端到端的方式进行训练。该模型的前馈不需要邻接矩阵,在计算训练过程中的损失时只是参考图结构,与传统的图形建模相比,具有更大的灵活性,因为该框架可以在没有完整图结构信息的情况下进行批量训练。

如算法1所示,在每个batch中,随机抽取 B 个节点,并取对应的邻接信息 \tilde{A} 和节点特征 X 。需要注意的是,对于某些节点 i ,由于批次采样的随机性,可能会出现批次中没有正样本的情况,在这种情况下,

将删除节点 i 的损失。

算法1 GMLP

输入 特征矩阵 X ;邻接矩阵的 r 次幂 B ;训练/验证/测试索引;待训练的模型参数 B, α, τ ;训练迭代次数 T

输出 优化模型参数 θ

1. $t \leftarrow 0$

2. while $t < T$ do

3. 对节点矩阵 $X[\text{Id}_B]$ 和邻接矩阵 $\tilde{A}[\text{Id}_B, \text{Id}_B]$ 进行特征采样

4. 计算损失

5. 执行反向传播,并更新模型参数 θ

6. 结束循环

在推理过程中,传统的图建模(如GCN)同时需要邻接矩阵和节点特征作为输入^[15],而本文基于MLP的方法只需要节点特征作为输入。因此,当邻接信息被破坏或丢失时,GMLP仍然可以提供一致可靠的结果。此外,在传统的图建模中,图信息被嵌入到输入的邻接矩阵中。对于这些模型,图节点变换的学习在很大程度上依赖于内部消息传递,而内部消息传递对每个邻接矩阵输入中的连接非常敏感^[16]。由于本文对图形结构的监督是建立在损失函数上的,因此模型能够在节点特征转换过程中学习图结构的分布,而无需前馈消息传递。

1.4 分布式异常流量检测单元

如图4所示,本文提出2个独立的模型,用于更新节点及其对应边的属性,基于此构建了部署在边缘转发器上的分布式检测单元。该架构的核心模块由边缘GMLP和节点GMLP组成,分别用于对节点和边的状态进行分类,更新节点及其相应边的属性。边缘检测单元用于特征分类并预测相邻节点上异常的概率,而节点检测单元用于节点的特征更新并计算导致其自身异常状态的概率。

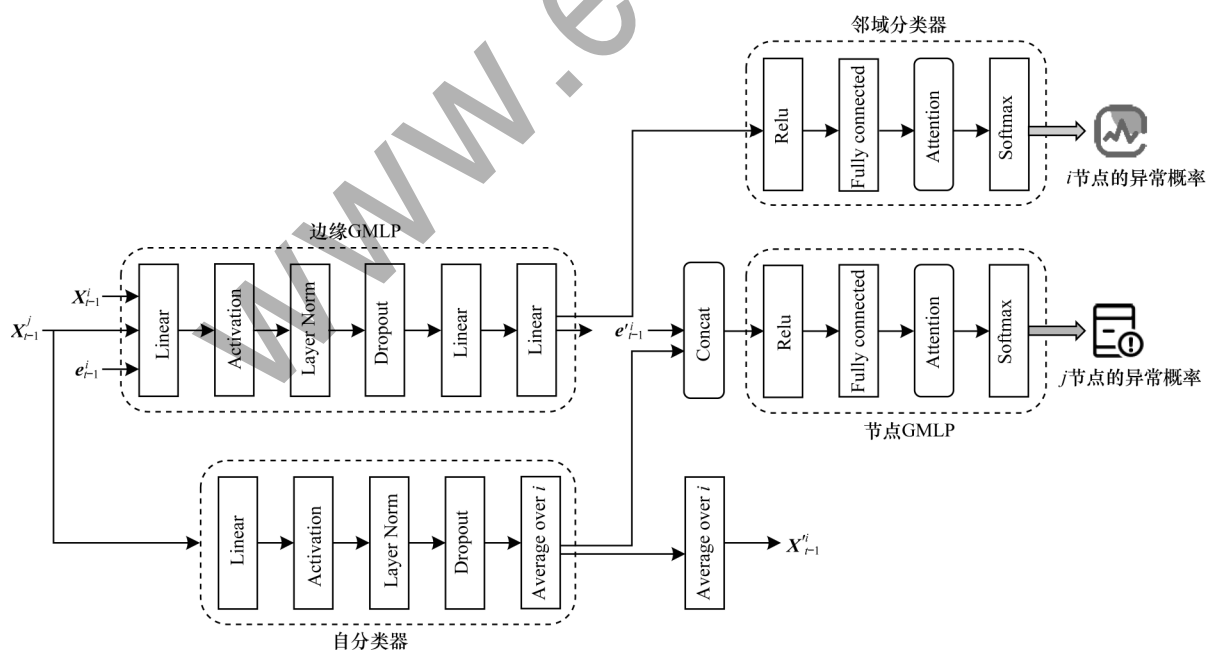


图4 单个异常流量检测单元结构

Fig.4 Structure of single traffic anomaly detection unit

与常规的图神经网络模型运算不同,本文实现了一个通信通道,在通道内建立了信息交换邻域,用于组合边缘GMLP和节点GMLP的信息。模型的输入分别代表边特征的3个属性和节点特征的5个属性,每个神经元通过单向链接连接。具体到细节,定义了输入和输出:假设有节点 j 及其相邻节点 $i=1,2,\dots,N$,边的输入由邻居对应的边缘特征向量(在 $t-1$ 时刻为 \mathbf{X}_{t-1}^i)、节点自身的信息(\mathbf{X}_{t-1}^j)以及邻居对应的边缘特征向量(\mathbf{e}_{t-1}^i)组成,将数据通过全连接层的输出更新边缘特征向量。与此同时,节点GMLP模块同样根据收集到的信息更新节点自身的特征表示,然后将更新的边缘特征向量与第 i 个节点的特征进行拼接作为Softmax分类器的输入,最后经过分类得到节点 j 的异常概率。与其他集中式入侵检测系统相比,这种实现信息交换的方式不需要显式的消息传递,有效地减少了对资源的占用。

此外,在图神经网络的正向传播过程中,需要关注起重要作用的节点信息,而忽略起次要作用的节点信息。为了进一步提高检测的精度,本文在最后一层分类之前加入了注意力机制模块,当每个节点更新隐含层的输出时,通过计算相邻节点的关注度来给每个相邻节点分配不同的权值,并将权值较高的节点作为神经网络关注的重点^[17]。注意力机制的引入减小了处理高维度数据的计算负担,使检测系统更专注于找到数据中显著的相关有用信息,从而提高输出质量。

2 实验结果与分析

2.1 实验环境

为了评估本文方案的检测性能,利用Python、NumPy、Pandas、Pytorch等工具,在64位计算机上使用Intel® i9-9700K 16 GB RAM, Nvidia GeForce RTX 2080Ti 32 GB和10.2版本的CUDA进行模拟实验。

2.2 实验数据

本文使用的数据集为CTU-13^[18],这是2011年在CTU大学捕获的僵尸网络流量数据集。该数据集包含13种不同的攻击,每个数据包都包含有关各种客户端和服务器的信息。网络由30个转发器和170个物联网设备组成,这些转发器和物联网设备根据CTU中的分布式设备交换数据。数据集的统计信息如表1所示。

表1 CTU-13数据集特征描述

Table 1 Feature description of CTU-13 dataset

特征	描述
Source IP	网络流量的源IP地址
Destination IP	网络流量的目的IP地址
Time stamp	时间戳
Total forward packets	转发方向的数据包总数
Total backward packets	反向数据包总数
Flow bytes	此IP上每秒字节数
Fwd packets	此IP上每秒转发方向数据包总数
Bwd packets	此IP上每秒反向平均数据包总数

2.3 蠕虫攻击模拟及数据集预处理

2.3.1 蠕虫攻击模拟

为了更好地描述受感染转发器上的传播特性并增加攻击的多样性,在数据集中模拟蠕虫攻击,其网络流量实例同样被标记为正常和异常。注入文献[16,19]的僵尸网络签名的异常流量,通过检查这种类型的特定攻击,能够更好地验证GNN在网络传播中检测攻击方面的优势。在图5中给出一个示例:边缘转发器的代理收到来自IoT的监控摄像头的蠕虫攻击,显示异常行为的恶意软件试图在物联网中进一步传播感染,将攻击带到转发器的相邻节点中,到第二阶段攻击已经蔓延到相邻的转发器。

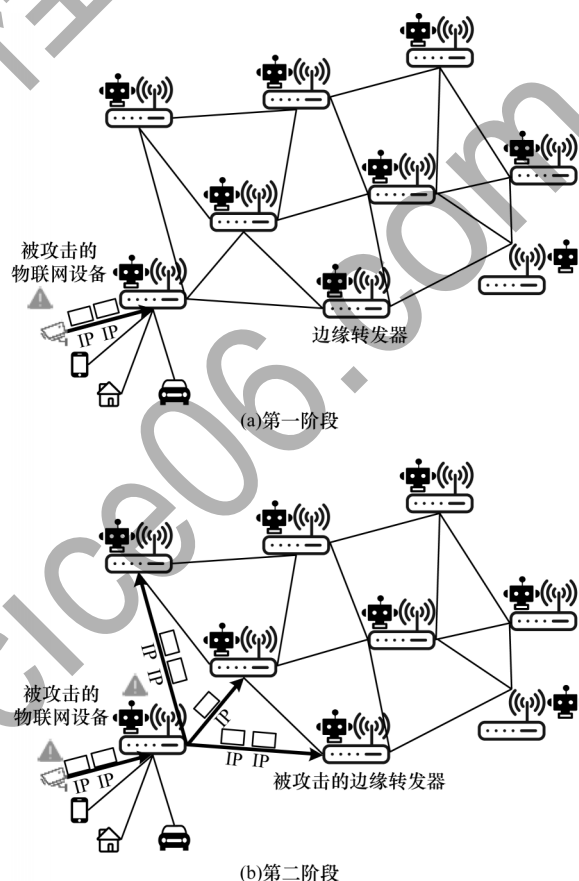


图5 蠕虫攻击两阶段过程

Fig.5 Process of two stages of worm attack

2.3.2 数据集预处理

CTU-13原始网络数据集包含多种类型的数据,因此,在模型训练和测试前需要进行预处理,以便相关特征能够归类到底层图的边和顶点。为了提取异常行为相关的特征,本文利用时间戳、源和目的IP地址、数据包数、字节数以及方向值的信息,数据集的信息交换主要根据源IP地址和目的IP地址以连续流的形式描述。为了在图的每个节点和边上特定地分配特征,对数据进行分组以提供每个节点的信息,从而分别为节点和边构造特征向量。

假设 j 是所关注的图, i 是指定时间窗口,定义一个包含多个索引记录的特征向量 $\mathbf{r}_i^j = [\mathbf{ps}_i, \mathbf{pr}_i, \mathbf{bs}_i, \mathbf{br}_i, \mathbf{dur}_i]$,其中, j 可以是节点 r_i^{node} 或边 r_i^{edge} 。该向量

包含以下记录:ps定义了在一定时间段内从一个节点/边缘发送到另一个节点/边缘的数据包数量;pr定义了接收的数据包数量;bs定义了在一定时间内从一个节点/边发送到另一个节点/边的字节数;br显示了从节点接收的字节数;dur(连接持续时间)描述了2个节点/边缘交换数据的连接时间和遵循日期时间格式的时间戳。最后, $\Delta t \in [t_1, t_2]$, $R_{\Delta t}$ 定义为 $R_{\Delta t} = \{r_i' | \text{timestamp} \in \Delta t\}$, N 定义为 $N = \text{length}(R_{\Delta t})$ 。表2和表3给出了从原始网络数据中提取的特征组(节点和边),以及相应的数学解释。

表2 节点特征矩阵描述

Table 2 Node characteristic description

特征	描述
平均发送的数据包总数	$f_{\text{aps}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{ps}_i}{N}, \text{ps} \in r_i^{\text{node}} \forall r_i^{\text{node}} \in R_{\Delta t}$
平均接收的数据包总数	$f_{\text{apr}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{pr}_i}{N}, \text{pr} \in r_i^{\text{node}} \forall r_i^{\text{node}} \in R_{\Delta t}$
平均发送的字节数	$f_{\text{abs}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{bs}_i}{N}, \text{bs} \in r_i^{\text{node}} \forall r_i^{\text{node}} \in R_{\Delta t}$
平均接收的字节数	$f_{\text{apr}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{br}_i}{N}, \text{br} \in r_i^{\text{node}} \forall r_i^{\text{node}} \in R_{\Delta t}$
平均连接持续时间	$f_{\text{acd}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{dur}_i}{N}, \text{dur} \in r_i^{\text{node}} \forall r_i^{\text{node}} \in R_{\Delta t}$

表3 边特征描述

Table 3 Edge characteristic description

特征	描述
平均发送的数据包总数	$f_{\text{aps}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{ps}_i}{N}, \text{ps} \in r_i^{\text{edge}} \forall r_i^{\text{edge}} \in R_{\Delta t}$
平均发送的字节数	$f_{\text{abs}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{bs}_i}{N}, \text{bs} \in r_i^{\text{edge}} \forall r_i^{\text{edge}} \in R_{\Delta t}$
平均连接持续时间	$f_{\text{acd}}^{\Delta t}(j) = \frac{\sum_{i=1}^N \text{dur}_i}{N}, \text{dur} \in r_i^{\text{edge}} \forall r_i^{\text{edge}} \in R_{\Delta t}$

2.4 评估指标

异常流量检测的性能指标依赖于混淆矩阵。在混淆矩阵中,真正类(T_p)为正确分类的异常流量实例;假正类(F_p)为错误分类的正常流量实例;真反类(T_n)为正确分类的正常流量实例;假反类(F_n)为错误分类的异常流量实例。这4项用于生成以下绩效评估指标:

准确率,即模型正确分类的样本数与样本总数的比值,计算公式如下:

$$A_{\text{accuracy}} = \frac{T_p + T_n}{T_p + F_n + F_p + T_n} \quad (13)$$

精确率,即模型正确分类的正常样本数与正常样本总数的比值,计算公式如下:

$$P_{\text{precision}} = \frac{T_p}{T_p + F_p} \quad (14)$$

召回率(也称为查全率),即模型正确分类的入侵样本数与正确分类的样本总数的比值,计算公式如下:

$$R_{\text{recall}} = \frac{T_p}{T_p + F_n} \quad (15)$$

误报率(False Positive Rate, FPR),即被误报为入侵的正常样本数与正常样本总数的比值,计算公式如下:

$$F_{\text{FPR}} = \frac{F_p}{F_p + T_n} \quad (16)$$

2.5 实验结果分析

2.5.1 检测性能比较

为证明本文方案的优势,使用经典的机器学习算法和深度学习方法在数据集CTU-13上进行实验对比,其中包括3种机器学习方法和3种深度学习方法。使用Adam优化器来训练模型,统一将批大小设置为1 024,学习率设置为0.000 1,批次设置为30,dropout设置为0.5。使用Pytorch构建检测模型,实验各项性能指标如表4所示。结果表明,本文方案在CTU-13数据集上准确率、精确率、召回率和误报率最高可达0.999 3、0.982 1、0.995 3和0.42,准确率较其他对比方案最高提升了0.269 2。这得益于本文改进的图神经网络模型在大型的数据集中能够更好地学习到复杂特征,因为越大的数据集中通信模式越复杂,IP节点和交互边越多,本文改进的模型相比于其他方法更容易发挥出优势,更准确地检测出异常流量。此外,注意力机制的引入也进一步提升了检测效果,使得本文方案检测性能优于其他方案。

表4 不同方案检测效果比较

Table 4 Comparison of detection effects of different schemes

方案	准确率	精确率	召回率	误报率
Decision Tree ^[20]	0.787 3	0.748 1	0.777 8	0.183 4
Naïve Bayes ^[21]	0.807 8	0.816 4	0.875 3	0.154 2
SVM ^[22]	0.856 4	0.901 7	0.921 5	0.076 3
RNN ^[23]	0.957 6	0.903 3	0.934 9	0.028 2
CNN ^[24]	0.918 3	0.920 6	0.950 6	0.016 0
CNN+LSTM ^[25]	0.971 5	0.980 4	0.969 4	0.008 6
本文方案	0.999 3	0.982 1	0.995 3	0.004 2

利用Matplotlib根据实验结果绘制ROC曲线图。定义横坐标为假阳性率,纵坐标为真阳性率,根据ROC曲线可以看到模型在任意阈值下的学习效果。ROC曲线将整个图形分为两部分,曲线以下的面积称为曲线下面积(AUC),表示检测的准确性,曲线越接近左上角性能越好,AUC值越大(曲线下面积越大),预测精度就越高^[26]。选取检测性能比较好的3个深度学习方法进行AUC值的对比。如图6所示,本文模型在CTU-13数据上AUC值达到了0.99,较其他模型最高可提升17.8%,这也从另外一个角度

表明,在物联网环境下,分布式异常检测方案能够获得比较理想的检测效果。

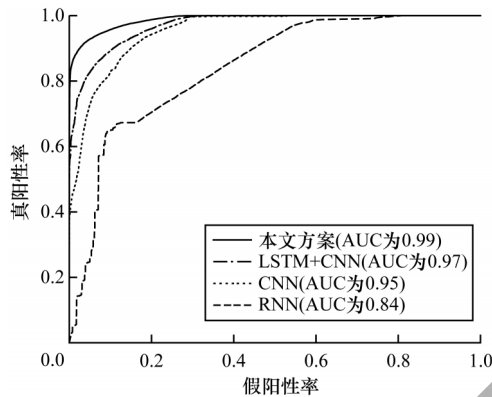


图6 不同方案的ROC曲线对比

Fig.6 Comparison of ROC curves of different schemes

2.5.2 资源和时间开销分析

为了验证部署分布式检测单元与集中式IDS相对的优势,本文还比较了不同方案在资源消耗和时间开销方面的性能。通过柱状图展示不同方案之间的资源消耗,如图7所示。根据数据不难看出,本文方案带宽消耗最低只有845 kb/s,与其他主流方法相比,资源消耗显著降低,这是因为分布式异常检测不需要将数据传输到云服务器的IDS进行计算,各个检测单元以较少的带宽占用降低了集中式IDS的资源消耗。

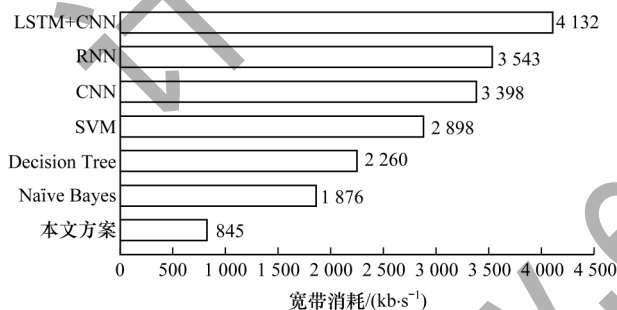


图7 不同方案的带宽消耗比较

Fig.7 Bandwidth comparison of different schemes

表5展示了本文方案和其他对比方案的训练参数数量和运行时间,使用GPU来加快所有模型的训练速度。可以看出,本文方案通过改进传统GCN的消息传递模块,在时间开销方面取得了良好的提升,训练时间和训练速度均有所减少。同时算法中的可训练参数较少,可以实现高效的并行计算。

表5 不同方案的计算复杂度比较

Table 5 Computational complexity comparison of different schemes

方案	可训练参数数量/ 10^3	训练时间/s	测试时间/s
RNN	18.6	168.3	10.40
CNN	23.2	172.5	12.50
LSTM+CNN	43.6	266.1	18.60
本文方案	6.3	98.3	9.26

2.5.3 消融实验分析

本节进行消融实验,验证不同改进点对流量异常检测的效果,实验结果如表6所示。其中,改进1表示替换GCN中的消息传递模块,使用GMLP来训练图神经网络;改进2表示注入蠕虫传播攻击,在各个转发器IP节点部署分布式异常检测单元;改进3表示引入注意力模块。分别从方案中删除几个改进点,组成新的模型。

表6 消融实验结果

Table 6 Ablation experimental results

方案	AUC	带宽消耗/($\text{kb}\cdot\text{s}^{-1}$)	测试时间/s
改进2+改进3	0.95	1 163	26.30
改进1+改进3	0.97	5 634	11.67
改进1+改进2	0.97	876	9.18
改进1+改进2+改进3	0.99	845	9.26

由表6中数据可知:改进的GMLP模块可以使检测时间降低一半,因为原本的图神经网络结构要不断更新相邻节点之间的消息传递,导致训练和测试时间较长;同时分布式异常检测单元实现了异常检测的本地化,避免了频繁与集中式IDS进行数据交互,带宽和资源消耗得以明显降低;此外,注意力机制使得模型在训练时更关注主要权重学习和优化重点参数,准确率和精度得到进一步提升。

3 结束语

本文结合物联网环境中设备节点复杂的特点以及低时延和高精度的检测需求,提出一种分布式异常流量检测方案。针对图卷积神经网络进行优化,以改进的多层感知机替换图学习中原有的消息传递模块,使模型更适用于物联网环境。在此基础上,结合物联网节点众多的特性,设计实现节点GMLP和边缘GMLP进行分布式流量异常检测,实现本地化的异常流量检测,并引入注意力机制进一步提升模型的检测效果。实验结果表明,本文方案不仅有效提高了检测精度,而且还减少了网络通信中的开销,加快了检测速度。下一步将对更多类型的流量数据集进行图结构分析,在更广泛的场景中进行模型训练和测试。

参考文献

- [1] 吴吉义,李文娟,曹健,等. 智能物联网AIoT研究综述[J]. 电信科学, 2021, 37(8): 1-17.
WU J Y, LI W J, CAO J, et al. AIoT: a taxonomy, review and future directions[J]. Telecommunications Science, 2021, 37(8): 1-17. (in Chinese)
- [2] RAHMAN M A, ASYHARI A T, LEONG L S, et al. Scalable machine learning-based intrusion detection system for IoT-enabled smart cities[J]. Sustainable Cities and Society, 2020, 61: 1-10.
- [3] QIU H, DONG T, ZHANG T W, et al. Adversarial attacks against network intrusion detection in IoT systems[J]. IEEE Internet of Things Journal, 2021, 8(13): 10327-10335.

- [4] FERRAG M A, MAGLARAS L, AHMIM A, et al. RDTIDS: rules and decision tree-based intrusion detection system for Internet-of-Things networks [J]. *Future Internet*, 2020, 12(3):44.
- [5] MA Y H, YANG Q, GAO Y J. An Internet of Things intrusion detection method based on CNN-FDC [C]// *Proceedings of 2021 International Conference on Intelligent Transportation, Big Data & Smart City*. Washington D. C. , USA: IEEE Press, 2021: 174-177.
- [6] KAUSHIK S. Enhanced the intrusion detection accuracy rate and performance using deep CNN-LSTM [D]. Dublin, Ireland: National College of Ireland, 2021.
- [7] KOZIK R, PAWLICKI M, CHORAŚ M. A new method of hybrid time window embedding with transformer-based traffic data classification in IoT-networked environment [J]. *Pattern Analysis and Applications*, 2021, 24(4): 1441-1449.
- [8] ASIF N A, SARKER Y, CHAKRABORTTY R K, et al. Graph neural network: a comprehensive review on non-euclidean space [J]. *IEEE Access*, 2021, 9: 60588-60606.
- [9] YAO Y P, SU L Y, ZHANG C, et al. Marrying graph kernel with deep neural network: a case study for network anomaly detection [C]// *Proceedings of International Conference on Computational Science*. Berlin, Germany: Springer, 2019: 102-115.
- [10] MAHMUD M R, RAMAMOCHANARAO K, BUYYA R. Application management in fog computing environments: a taxonomy, review and future directions [J]. *ACM Computing Surveys*, 2020, 53(4): 1-43.
- [11] HASSAN N, YAU K L A, WU C. Edge computing in 5G: a review [J]. *IEEE Access*, 2019, 7: 127276-127289.
- [12] MANSOURI Y, BABAR M A. A review of edge computing: features and resource virtualization [J]. *Journal of Parallel and Distributed Computing*, 2021, 150: 155-183.
- [13] CHIANG W L, LIU X Q, SI S, et al. Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks [C]// *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, USA: ACM Press, 2019: 257-266.
- [14] ZHANG C, PAN X, LI H P, et al. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification [J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018, 140: 133-144.
- [15] LI G H, MULLER M, THABET A, et al. DeepGCNs: can GCNs go as deep as CNNs? [C]// *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision*. Washington D. C. , USA: IEEE Press, 2019: 9267-9276.
- [16] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks [EB/OL]. [2021-10-11]. <https://arxiv.org/abs/1609.02907>.
- [17] FUKUI H, HIRAKAWA T, YAMASHITA T, et al. Attention branch network: learning of attention mechanism for visual explanation [C]// *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Washington D. C. , USA: IEEE Press, 2019: 10705-10714.
- [18] GARCÍA S, GRILL M, STIBOREK J, et al. An empirical comparison of botnet detection methods [J]. *Computers & Security*, 2014, 45: 100-123.
- [19] KOLIAS C, KAMBOURAKIS G, STAVROU A, et al. DDoS in the IoT: Mirai and other botnets [J]. *Computer*, 2017, 50(7): 80-84.
- [20] PENG K, LEUNG V C M, ZHENG L X, et al. Intrusion detection system based on decision tree over big data in fog environment [J]. *Wireless Communications and Mobile Computing*, 2018, 2018: 1-10.
- [21] MUGHAL M O, KIM S. Signal classification and jamming detection in wide-band radios using Naïve Bayes classifier [J]. *IEEE Communications Letters*, 2018, 22(7): 1398-1401.
- [22] REDDY R R, RAMADEVI Y, SUNITHA K V N. Effective discriminant function for intrusion detection using SVM [C]// *Proceedings of 2016 International Conference on Advances in Computing, Communications and Informatics*. Washington D. C. , USA: IEEE Press, 2016: 1148-1153.
- [23] PARK S H, PARK H J, CHOI Y J. RNN-based prediction for network intrusion detection [C]// *Proceedings of 2020 International Conference on Artificial Intelligence in Information and Communication*. Washington D. C. , USA: IEEE Press, 2020: 572-574.
- [24] KIM J, KIM J, KIM H, et al. CNN-based network intrusion detection against denial-of-service attacks [J]. *Electronics*, 2020, 9(6): 916.
- [25] JIANG K Y, WANG W Y, WANG A L, et al. Network intrusion detection combined hybrid sampling with deep hierarchical network [J]. *IEEE Access*, 2020, 8: 32464-32476.
- [26] WANG Z, MARTIN R. Model-free posterior inference on the area under the receiver operating characteristic curve [J]. *Journal of Statistical Planning and Inference*, 2020, 209: 174-186.

编辑 金胡考