软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2020,31(3):748−762 [doi: 10.13328/j.cnki.jos.005903]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于图神经网络的动态网络异常检测算法*

郭嘉琰[1], 李荣华[2], 张 岩[1], 王国仁[2]

[1](北京大学 信息科学技术学院,北京 100871)

[2](北京理工大学 计算机学院,北京 100081)

通讯作者: 李荣华, E-mail: lironghuabit@126.com

**摘 要**: 动态变化的图数据在现实应用中广泛存在,有效地对动态网络异常数据进行挖掘,具有重要的科学价值和实践意义.大多数传统的动态网络异常检测算法主要关注于网络结构的异常,而忽视了节点和边的属性以及网络变化的作用.提出一种基于图神经网络的异常检测算法,将图结构、属性以及动态变化的信息引入模型中,来学习进行异常检测的表示向量.具体地,改进图上无监督的图神经网络框架 DGI,提出一种面向动态网络无监督表示学习算法 Dynamic-DGI.该方法能够同时提取网络本身的异常特性以及网络变化的异常特性,用于表示向量的学习.实验结果表明,使用该算法学得的网络表示向量进行异常检测,得到的结果优于最新的子图异常检测算法 SpotLight,并且显著优于传统的网络表示学习算法.除了能够提升异常检测的准确度,该算法也能够挖掘网络中存在的有实际意义的异常.

**关键词**: 动态网络异常检测;图神经网络;图深度学习

**中图法分类号**: TP18

中文引用格式: 郭嘉琰,李荣华,张岩,王国仁.基于图神经网络的动态网络异常检测算法.软件学报,2020,31(3):748−762. http://www.jos.org.cn/1000-9825/5903.htm

英文引用格式: Guo JY, Li RH, Zhang Y, Wang GR. Graph neural network based anomaly detection in dynamic networks. Ruan Jian Xue Bao/Journal of Software, 2020,31(3):748−762 (in Chinese). http://www.jos.org.cn/1000-9825/5903.htm

## Graph Neural Network Based Anomaly Detection in Dynamic Networks

GUO Jia-Yan[1], LI Rong-Hua[2], ZHANG Yan[1], WANG Guo-Ren[2]

[1](School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

[2](School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

**Abstract**: Dynamic graph structured data is ubiquitous in real-life applications. Mining outliers on dynamic networks is an important problem, which is very useful for many practical applications. Most traditional network outlier detection algorithms focus mainly on the strutraulal anomaly, ignoring the nodes and edges' attributes, and the time-varying features as well. This study proposes a graph neural network based network anomaly detection algorithm which can capture the nodes and edges' attributes and time-varying features and fully uses these features to learn a representation vector for each node. Specifically, the proposed algorithm improves an unsupervised graph neural network framework called DGI. Based on DGI, a new danamic DGI algorithm is proposed, which is called Dynamic-DGI, for dynamic networks. Dynamic-DGI can simultaneously extracts the abnormal characteristics of the network itself and the abnormal characteristics of the network changes. The experimental results show that the proposed algorithm is better than the state-of-the-art

* 基金项目: 国家自然科学基金(61772346, U1809206, 61532001, 61332006, 61332014, 61328202, U1401256); 教育部-中国移动科研基金(MCM20170503)

Foundation item: National Natural Science Foundation of China (61772346, U1809206, 61532001, 61332006, 61332014, 61328202, U1401256); China MOE and China Mobile Joint Research Foundation (MCM20170503)

本文由人工智能赋能的数据管理、分析与系统专刊特约编辑李战怀教授、于戈教授和杨晓春教授推荐.

收稿时间: 2019-07-19; 修改时间: 2019-09-10, 2019-11-25; 采用时间: 2019-12-18; jos 在线出版时间: 2020-01-10

CNKI 网络优先出版: 2020-01-10 14:29:55, http://kns.cnki.net/kcms/detail/11.2560.TP.20200110.1429.016.html

anomaly detection algorithm SpotLight, and is significantly better than the traditional network representation learning algorithms. In addition to improving the accuracy, the proposed algorithmis also able to mine interesting anomalies in the network.

**Key words**: anomaly detection in dynamic network; graph neural network; deep learning on graphs

网络结构数据(图)因为其强大的表示能力,在过去的几年间得到广泛关注.现实生活中的网络分为静态网络和动态网络:静态网络可以理解为不随时间进行任何变化的网络,比如某个时间点上某城市的交通网络;相比于静态网络,动态变化的网络形式在现实世界中更加普遍,比如社交网络、账户之间的转账交易网络以及计算机通信网络等[1].在这些随时变化的网络中可能存在一些元素,其变化规律或特征因与一般的元素不同而表现出异常的行为,比如计算机网络中具有攻击行为的通信[2]、社交网络中的虚假信息传播[3]以及学术合著网络中不同领域学者之间突然的合作[4]等.尽早地挖掘网络中存在的这些异常,对于维护社会稳定、防御网络攻击或发现新兴的交叉学科方向具有重要的意义[5−7].

如何在动态网络中挖掘异常元素是比较困难的问题.动态网络主要有以下一些特点:1) 网络结构处于不确定的变化之中,每一时刻都有新的点或边加入或删除;2) 网络的属性处于不确定的变化之中,同一节点或边在不同时刻的属性特征可能不同.这些特点导致我们不能够使用传统静态网络上的异常检测算法来解决该问题.同时,网络中的异常包括有节点的异常、边的异常以及子图的异常,这些不同的异常形式又给图上的异常检测增添了复杂性[8].传统方法主要关注于网络的结构特征,通过找寻结构变化的异常来探测异常元素.值得注意的是,网络元素除具有结构特征外,也具有属性特征,要解决更加一般意义上的图数据的异常检测问题,就必须同时考虑图中元素的结构和属性;其次,各方面对于“异常”的定义十分模糊,目前还没有比较统一的定义形式.总结而言,目前动态网络异常检测存在的主要问题有:

- 如何同时结合图的结构特征和属性特征来更好地挖掘异常.图上元素除了因结构产生的异常外,其本身具有的属性也可能使其具有不同于一般元素的特性,需要找到合适的方法,结合两方面的信息来确定异常;
- 带有标注的动态网络异常数据很少,异常数据和正常数据的样本数非常不平衡.如何使用无监督的方式来获得动态网络的表示,并在此基础上进行异常元素的挖掘;
- 动态网络的变化特征.动态网络的动态性通常表现为结构的变化和属性的变化,异常性也包括元素本身的异常以及变化的异常.如何将这两者同时编码进表示向量,是一个需要解决的问题.

为了解决特征表示的问题,本文引入图的表示学习技术.表示学习是随着深度学习的出现而逐渐发展起来的,最经典的图上表示学习技术可以追溯到 2014 年 Perozzsi 等人[9]提出的 Deepwalk.学得的网络表示含有很多有用的信息,比如越相似的节点,其表示向量之间的距离越小等,这为下游的机器学习任务提供了良好的输入特征.对于异常检测问题,可以设计特殊的表示学习方法来让表示向量包含探测元素一般性的特征(结构特征和属性特征),之后,再使用比较成熟的数据流上的异常检测算法——鲁棒随机切割森林(robust random cut forest,简称 RRCF)[10]等挖掘出具有异常的元素.值得注意的是:图上经典的表示学习技术都属于转导学习(transductive learning)的范畴,当有新的数据到来时,必须让模型重新进行学习,才能获得新数据的表示.而对于动态变化的网络来说,因为内存和时间上的限制,不能重新进行训练,要求模型对新的数据直接给出其表示,这就需要使用归纳学习(inductive learning)的方式.为了解决该问题,本文引入图神经网络(graph neural network,简称 GNN)[11−13],通过利用 GNN 作为图的特征抽取器来提取图结构和属性中有用的特征,并应用到更深层的神经网络中.

为了将图的变化作为一个特征编码来探测图的变化上的异常,本文使用门控的循环神经网络 LSTM 来对图的变化进行建模.使用 LSTM 的好处是可以解决长期依赖[14]以及梯度消失的问题,对较长的序列处理比较有利.本文使用 LSTM 将变化的信息编码,并结合 GNN 提取到的整个网络本身的属性和结构特征一起编码进表示向量,从而作为网络的表示.之后,为了进行无监督表示学习,本文扩展 Deep Graph Infomax 的无监督表示学习方法,并提出 Dynamic-DGI(dynamic deep graph infomax)的动态网络无监督表示学习框架.

综上所述,本文的主要贡献包括:将图神经网络应用于动态网络异常检测,从而使网络异常检测可以同时抓住结构上的异常以及属性上的异常.提出 Dynamic-DGI 的时序网络表示学习框架,从而使模型能够脱离标记数

据来学习网络变化的一般特征.在多个数据集上检测了本文的方法和对比方法.

## 1 相关工作

### 1.1 动态网络异常检测

Aggarwal 等人[15]最先关注于图流上的异常检测,他们认为图流上的异常是连接不同紧密区域的边,由此提出了用结构连通模型建模动态网络中的边的方法.具体做法是:维持一个当前节点集合的划分,并利用数据流的采样让划分成的不同子图内部尽可能紧密.这样,当新的边到来时,就能利用边在不同子图间的信息来为边进行异常值(anomaly score)的打分.这种做法能够将连接不相交紧密区域间的边作为异常找出来,而缺点是需要提前知道所有点的信息来启动划分,并且为了提升运算精度,必须维持多个模型同时进行计算.

Ranshous 等人[16]关注于边流上的异常检测,将动态网络建模成随时间不断到来的边,并在其基础上检测异常的边.为了定义异常性,他们设置了 3 个经验性的异常指标——采样分数(sample score)、偏好依附分数(preferential attachment score)和同质性分数(homophily score)来为流中的边进行打分.这 3 种指标都是建立在网络结构之上的.为了维持并更新网络结构信息,提出了一种多维的 CM-Sketch(count min sketch)的方法来维持边的计数,并对 3 个分数进行估计来得到最终的异常分数.这种做法可以快速查找结构上异常的边,缺点是使用了经验性的指标从而使方法的泛化能力降低.Eswaran 等人[2]将异常边认为是突发情况下出现的连接稀疏连接区域的边.由此,他们将加边之前与加边之后边周围节点之间的连通度进行对比,来对让连通度增大更多的边具有更高的异常值.该方法关注于突发情况中的边,因此更适合对网络攻击中的情况进行异常检测.缺点是维持 sample 时需要较多额外的空间,并且不能对一般情况中的异常进行检测.当异常并不存在于突发情况中时,这种做法就不能够十分有效.

Manzoor 等人[17]关注于异质图(heterogeneous graph)流上的异常检测.异质图是一种点(边)可以有多种存在形式的图,比如知识图谱(knowledge graph)等.Manzoor 的具体做法是:将一定时间段内的边构成的图表示成 *k*-shingle 的形式,并利用多个流哈希函数(StreamHash)对图进行 sketch.之后,使用流上的聚类算法来将得到的 sketch 向量进行聚类,并为图的异常程度进行打分.这种方法的时间和空间损耗很少,缺点是只能适用于异质图的情况,并且在确定初始聚类中心时需要使用一些边进行启动.Eswaran[18]将 sketch 技术应用到多图中,并将动态网络中的异常定义为突然出现或消失的稠密子图.这种类型的异常在网络攻击中十分常见,比如拒绝服务攻击(denial of service)等.他们的做法是:用哈希函数对某一时刻图中的边进行采样并投射到 sketch 空间,再使用流上的异常检测算法 RRCF 等[10]来对得到的 sketch 向量进行异常检测.这种方法可以很容易扩展到二部图以及边上的异常检测等.缺点是只能根据结构探测异常的稠密子图,因此只适用于特定的情境.

Yu 等人[19]首先将深度学习技术应用于动态网络异常检测中.他们首先使用随机游走来产生节点的上下文,并将节点的上下文的独热编码产生的矩阵输入到稀疏自编码器中(sparse autoencoder)来获得压缩后的节点向量表示,之后使用 Clique-Embedding 的损失函数来让上下文中的节点在表示空间中的距离尽可能小.这种方法具有较强的泛化能力,同时将异常分数定义为边到离其最近聚类中心的距离,而使其不受经验性指标的限制.但是随着网络的动态变化,网络结构特征也在发生变化,因此在处理一定数量的边之后就必须对模型进行更新,这就损耗了时间并影响了模型效果.

### 1.2 图神经网络

图神经网络是为了在图结构的数据上进行深度学习而发展出来的,在很多方向都发挥了重要的作用[20−22].其中,使用最为广泛的图神经网络是图卷积神经网络(graph convolutional network,简称 GCN)[23−25]和图注意力网络[26]等.

图神经网络自从提出以来主要被应用于节点分类等半监督学习任务.为了让其适用于无监督学习的情景,Hamilton 等人[27]提出使用类似 skip-gram 方法的损失函数作为学得的表示向量的约束.这种方法只考虑了节点的局部信息而不能使节点很好地利用全图信息.受 Hjelm 等人[28]提出的通过最大化全局特征抽象表示与局部

特征表示之间互信息(mutual information)来学习图像表示向量的启发,Velikovi 等人[29]提出了图上的无监督表示学习框架 DGI(DeepGraphInfoMax).本文将该框架应用于归纳学习以及动态网络的设置中,并专注于网络异常检测的场景设计了最大距离的读取函数.最大距离读取函数通过选取所有节点表示中与模型当前状态每一维的相差最大的值作为图的表示,从而凸显出图中最显著的特征,这对于进行异常检测十分有帮助.

Weber 等人[30]为探测金融交易网络中的异常提出了 EvolveGCN,该方法使用 GCN 作为图的特征提取器,并使用循环神经网络来建模网络的变化过程.但是该方法只适用于监督学习的模式,并且只使用网络中影响力最大的 Top-$k$ 节点代表某时刻网络的全部信息而忽略了网络的总体特征.本文的方法可以在网络总体信息的基础之上使用无监督的方式进行模型的学习.

## 2 基于图神经网络的动态网络异常检测

我们首先进行动态网路异常检测问题的定义:给定图流 $G=(G_{t_1},G_{t_2},...,G_{t_x})$,一个表示函数 $f_{embed}$ 和一个异常评分函数 $f_{score}$,要探测的异常图的集合 $G'\subseteq G$ 为 $\forall G_k\in G',f_{score}(f_{embed}(G_t)|\{f_{embed}(G_t)|t<k\})>c$.其中,$c$ 为一个常数,$c>0$.由此可知主要任务有 3 点:(1) 找到好的向量表示来体现图的整体特征;(2) 使模型能够记忆之前存在过的图的信息;(3) 找到合适的算法来给每一时刻的图进行异常打分,并认为异常分数大于阈值的图为异常图.

结合以上定义,本文提出一种使用图神经网络来进行动态网络异常检测的算法.该算法首先使用图神经网络将 $t$ 时刻的网络元素信息(节点、边)提取到特征空间,之后使用图上的无监督表示学习算法 DGI 将当前时刻的整个网络表示成一维的向量.在图的表示向量的基础上,使用成熟的流上的异常检测算法 RRCF 等为每一时刻的图进行打分,获取其异常分数.为了确定异常图,可以设定一个阈值并认为分数超过阈值的图存在异常.在进行网络表示学习的过程中,我们使用全局表示与局部表示互信息最大化的策略来进行图的表示学习.为了使模型能够利用每一时刻的图信息,我们使用 LSTM 来获取每一时刻网络全局表示的变化信息并加以处理.图 1 展示了本文算法的总体框架.

(1) 图的属性特征、结构特征的提取.使用图神经网络来提取某时刻图的属性特征和结构特征;

(2) 图的时间变化特征的提取.使用长短路记忆模型来结合不同时刻图的信息提取图的变化特征;

(3) 动态网络表示学习.使用最大化局部与全局表示互信息的策略来进行图表示向量的学习;

(4) 流数据的异常检测.使用数据流上的异常检测算法给出异常分数.

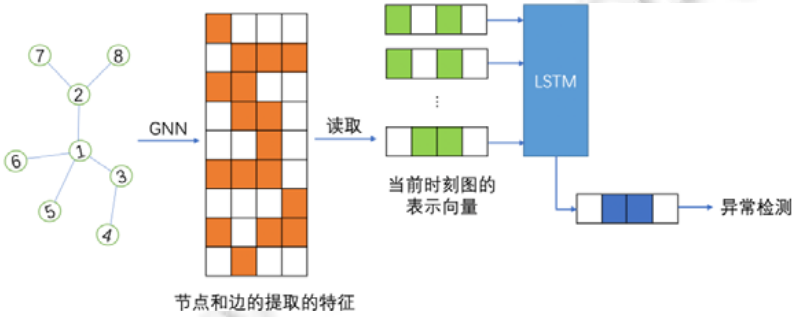



Fig.1 Dynamic network anomaly detection framework

图 1 动态网络异常检测算法框架

### 2.1 基于图神经网络的图特征提取

图神经网络因在图上的深度学习中发挥重要作用而成为近年来研究的热点,其本质是信息的传递和汇聚.给定图 $G=(V,E)$,其中,$V$ 是节点的集合,$E$ 是节点之间边的集合.令 $A$ 为 $G$ 的邻接矩阵,则图神经网络的一层操作可以分为节点信息传播和信息拼接两个步骤,如公式(1)和公式(2)所示:

$$h_{u,nei}^{l}=aggregate_{l+1}(\{h_v^l\,|\,v\in Neighbor(u)\}) \tag{1}$$

$$h_u^{l+1} = combine_{l+1}(h_u^l \| h_{u,nei}^l) \tag{2}$$

其中,$h_u^l$ 为第 $l$ 层节点 $u$ 的隐含表示,$h_{u,nei}^l$ 为第 $l$ 层 $u$ 的邻居信息的汇聚,$aggregate_l(\cdot)$和 $combine(\cdot)$分别为第 $l$ 层的聚合操作和更新操作.

在实际操作中,我们使用比较常用的图卷积神经网络来进行网络特征的提取.图卷积网络模仿图像上频率域的卷积操作,首先将图映射到频率空间,在频率空间进行卷积操作之后,再将其转换回节点空间.使用最多的图卷积神经网络由 Kipf 等人[25]提出,其一层操作为

$$Z^{(l+1)} = Act(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}Z^lW^l) \tag{3}$$

其中,$Z^l$ 为第 $l$ 层节点的隐含表示,$\tilde{A} = A + I$ 为加入了节点到自身环路的邻接矩阵,$\tilde{D} = \sum_{i=1}^{N}\tilde{A}_{ii}$ 为 $\tilde{A}$ 的度数矩阵,$Act(\cdot)$为激活函数.值得注意的是,在动态网络数据中,有的情况下,除了节点具有实际意义和属性外,边也具有实际的意义和属性,比如通信网络中两个 IP 地址之间建立的连接.因此在设计网络结构、属性特征提取器的时候,要同时考虑边的信息和节点的信息.本文的方法通过将图转换成对应的线图(line graph)来获取以边为基本元素的网络,其转换规则见公式(4):

$$E_{ij} = \begin{cases} 1, & e_{i,from} = e_{j,from} \text{ or } e_{i,to} = e_{j,to} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

其中,$e_{i,from}$ 为边 $i$ 的源节点,$e_{i,to}$ 为边 $i$ 的目标节点.则对应的线图上的特征提取网络为

$$Z_E^{(l+1)} = Act(\tilde{D}_E^{-1/2}\tilde{E}\tilde{D}_E^{-1/2}Z_E^lW_E^l) \tag{5}$$

使用两组图卷积神经网络结合 JK Network[31]的网络构造分别从原图和其对应的线图中提取特征并加以整合,可以得到如图 2 所示的图特征提取框架.在进行两部分信息的提取之后,该框架将这两部分信息进行拼接并做一个线性变换,从而获得所有节点和边的隐含表示.

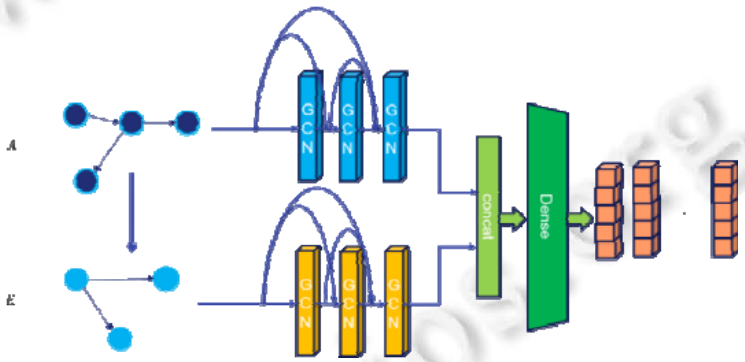



Fig.2 Graph feature extraction framework

图 2 图特征提取框架

### 2.2 基于互信息最大化的网络表示学习

本文使用最大化全局表示向量和局部表示向量之间互信息的方式来进行表示向量的学习.该思想最早来源于 Hjelm 等人[28]提出的进行图像表示学习的方法,Veličković 等人[29]将其扩展到图的情况中,并利用这种方法学习节点的表示向量.我们将这种方法扩展到学习图的全局表示向量的设置中,通过一个读取函数从节点和边的表示向量中获得图的全局表示,再用最大化互信息的做法进行全局表示向量互信息和局部表示向量互信息的最大化训练.为了使模型更好地抓住子图中最异常的特征,本文提出一种贪心读取的方法.该方法利用当前状态的信息对数据流中的边进行采样,使得越可能异常的边的信息进入模型越多.首先定义当前状态为 $C_t \in \mathbb{R}^d$,其中,$d$ 为表示向量的维度.令 $D:\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ 为两表示向量之间的距离,比如欧几里得距离、余弦距离等,则边的每一维的读取优先度为 $p(e_i) = x^{D(encoder(e_i),C_t)}$,其中,$x \in uniform(0,1)$,$encoder(\cdot)$为边空间到表示空间的函

数.该方法可以将每一维中与当前状态相差最多的信息读取出来,从而能够使当前最异常的信息流入图的表示空间中.整个表示学习的过程如图 3 所示.

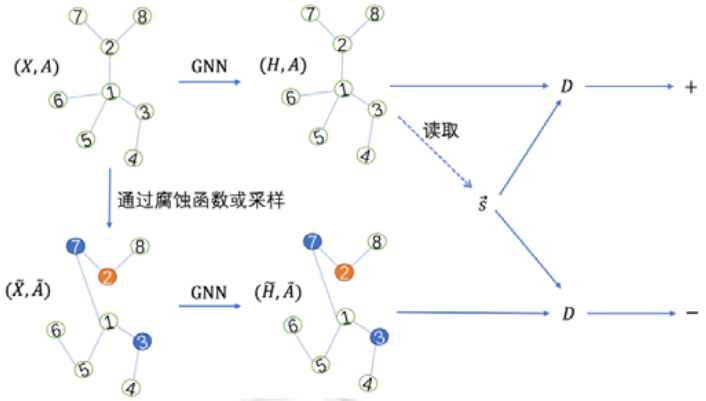



Fig.3 Network representation learning by mutual information maximization

图 3 互信息最大化网络表示学习

其中,左下方的图可以使用腐蚀函数从原图中获得或者从已经存在过的图中采样获得.$H_{\text{true}}$ 为从原图提取获得特征的隐含表示;$H_{\text{false}}$ 为从腐蚀过的或采样得到的图提取获得的特征的隐含表示;$S$ 是使用读取函数从原图的特征隐含表示中获得的全图的总结表示;$D$ 为一个判别器,用来使用全局表示来分别给正例和负例进行打分,通过给正例尽可能打高分并给负例打低分来进行图的表示向量的学习.最终的损失函数见公式(6):

$$L_1=\frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(X,A)}[\log\mathcal{D}(\vec{h}_i,\vec{s})]+\sum_{j=1}^{M}\mathbb{E}_{(\tilde{X},\tilde{A})}[\log(1-\mathcal{D}(\vec{\tilde{h}}_j,\vec{s}))]\right) \tag{6}$$

可以看出,公式(6)和生成对抗网络(GAN)的损失函数相类似,但是有本质的不同:GAN 中的负例是通过生成器从噪声中生成出的,而我们的算法的负例是通过对原图进行腐蚀或者从已经存在过的图中进行采样得到的.这种表示学习方法能够增大模型的全局表示和局部表示之间的互信息,从而使全局表示具有局部表示中比较一般的特征,也就意味着获得了比较能够体现网络整体性的表示向量.

### 2.3 基于长短路记忆模型的动态网络表示学习框架

本节介绍本文提出的动态网络表示学习框架 Dynamic-DGI.该方法结合 LSTM 和互信息最大化算法来进行动态网络的表示学习.LSTM 模型通常用来做序列建模,并能解决长序列建模可能带来的梯度消失和长期依赖问题.我们使用长短路记忆循环神经网络来对序列的变化进行建模.结合第 2.2 节中提到的最大化全局互信息和局部互信息的网络表示学习方法,本文总的模型框架如图 4 所示.该框架使用长短路记忆网络来提取网络变化的特征,并结合图神经网络提取到的结构、属性特征来形成当前时刻整个子图的表示向量.

其中,$\varepsilon$表示图提取器(图神经网络),$\mathcal{R}$表示读取函数,其读取全部元素的表示向量 $U_v\in\mathbb{R}^{n\times d}$ 并得到图的总结表示向量 $s_t\in\mathbb{R}^d$.假设在 $t$ 时刻有子图 $G_t=(X_t,A_t)$到来,首先使用图神经网络获得其结构、属性特征,并使用读取函数获得其全局表示 $s_t$;之后,将 $s_t$ 作为 $t$ 时刻的输入送入长短路记忆网络中来获得加入变化信息后的向量表示.在进行模型训练的过程中,加入变化损失式(7)来约束 LSTM 的特征提取:

$$L_2=\left\|y_t-\frac{1}{t-1}\sum_{i=1}^{t-1}y_i\right\|_2 \tag{7}$$

该损失函数使模型最终获得的加入时序信息的表示向量能够尽可能和之前所有的向量的均值相接近.使用这种方法的原因在于我们假设模型是在完全没有异常信息的数据上进行训练的.这样,当进行预测任务时,突然出现的异常子图的表示向量和其他时刻的子图的表示向量会具有较大的差距.结合 $L_1$ 和 $L_2$,可以得到模型的总的损失函数:

$$L=\alpha\frac{1}{N+M}(\sum_{i=1}^{N}\mathbb{E}_{(X_t,A_t)}[\log\mathcal{D}(\vec{h}_i,\vec{s})]+\sum_{j=1}^{M}\mathbb{E}_{(\tilde{X}_t,\tilde{A}_t)}[\log(1-\mathcal{D}(\vec{\tilde{h}}_j,\vec{s}_t))])+\beta\left\|y_t-\frac{1}{t-1}\sum_{i=1}^{t-1}y_i\right\|_2 \tag{8}$$

其中,$\alpha$和$\beta$为超参数.

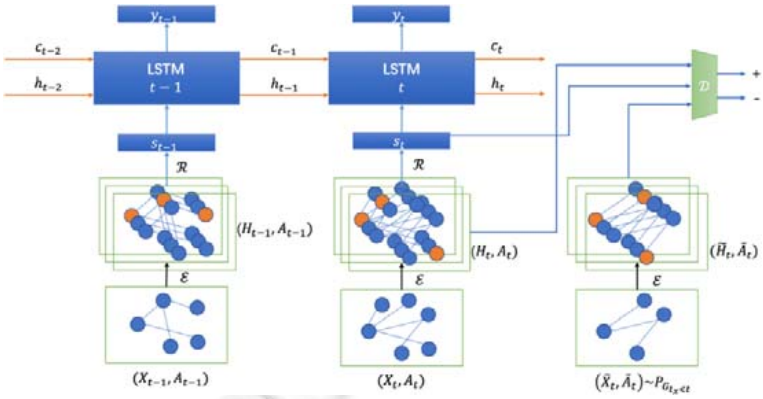



Fig.4 Dynamic representation learning framework

图 4 动态网络表示学习框架

### 2.4 数据流上的异常检测算法

#### 2.4.1 鲁棒随机切割森林

在使用动态网络表示学习框架学得表示向量之后,我们使用流上的异常检测算法检测异常.本文主要使用鲁棒随机切割森林算法来对表示向量进行处理.RRCF 算法主要基于两个观察:(1) 将异常点从所有的数据中区分出来比较容易;(2) 去掉异常点之后区分剩下的数据将比较困难.基于这两个性质,该算法结合鲁棒随机划分森林的数据结构来进行异常检测.RRCF 由一组 RRCT(robust random cut tree)组成,一个 RRCT 由以下方式定义:

**定义 1(RRCT)**. 点集 $S$ 上的一个鲁棒随机划分森林由以下步骤产生.

1. 选择一个随机的正比于 $\frac{l_i}{\sum_j l_j}$ 的维度,其中,$l_i=\max_{x\in S}-\min_{x\in S}$;
2. 选择 $X_i\sim Uniform[\min_{x\in S}x_i,\max_{x\in S}x_i]$;
3. 令 $S_1=\{x|x\in S,x_i\leqslant X_i\}$,并且 $S_2=S\backslash S_1$;
4. 在 $S_1$ 和 $S_2$ 上重复步骤 1~步骤 3,直到划分结果为单独的点.

由定义 1 的过程可以得到一棵 RRCT,多个 RRCT 就组成了 RRCF.值得注意的是:RRCT 为二叉树,由此可以构建一棵二叉哈夫曼树,并在树上定义每一个叶子结点的编码,而叶子结点的编码长度为其所在的树的位置的深度.令 $Z$ 为数据点的集合,$f(y,Z,T)$为树 $T$ 中点 $y$ 的编码长度(所在深度),则一个树的编码复杂度可以定义为所有数据点的编码复杂度的和:

$$|M(T')|=\sum_{y\in Z-\{x\}}f(y,Z,T) \tag{9}$$

之后定义数据点的异常分数为删去该异常点后复杂度的减少的期望:

$$score_{anomay}=E_{T(Z)}||M(T)||-E_{T(Z-\{x\})}||M(T(Z-\{x\}))|| \tag{10}$$

在实际使用时,为了防止出现串谋者(colloders)的干扰,可以使用特定采样的一组点而不是一个点来定义异常值.这样可以进一步增加算法的鲁棒性.

#### 2.4.2 Streaming *k*-means

使用流上的聚类算法时需要考虑的重要问题是:当一个新的数据点到来时,很难判断其是一个异常还是一个新的聚类中心.这时就可以根据节点到其最近的聚类中心的距离作为评价异常分数的标准,并同时更新聚类中心.Streaming *k*-means 就是动态更新聚类中心的一种聚类算法,其使用延迟系数(decayfactor)来动态地

更新聚类中心.令 $\{x_i\}_{i=1}^{n_0}$ 为已经存在的 $n_0$ 个数据点,此时在时间节点 $t'$有 $n'$个新的数据 $\{x_i'\}_{i=1}^{n'}$ 到来,新的聚类中心为 $\boldsymbol{c}$,延迟系数为$\alpha$,则对应的聚类中心更新为

$$c=\frac{\alpha c_0 n_0+(1-\alpha)\sum_{i=1}^{n'}x_i'}{\alpha n_0+(1-\alpha)n'} \tag{11}$$

之后定义异常分数为数据点到离其最近的聚类中心的距离:

$$score_{anomaly}=\|c_{nearest}-x_i\|_2 \tag{12}$$

这样做不需要提前设计经验性的指标,且当新的数据来时,不需要判断其是新的聚类中心还是真正的异常.

2.4.3 Encoder-Decoder-Encoder

该方法主要是为模型完全在正常数据上训练而设计的.本文的模型在学习时只在正常的数据上进行学习,因此可以使用这种架构的异常检测器来帮助完成异常检测.首先,Encoder-Decoder-Encoder 框架中的第 1 个 Encoder 是将实体编码为分布式向量的函数(神经网络).当信息被编码为分布式向量后,我们利用分布式向量训练一个 Decoder,使该分布式向量能够很好地吸收原信息中最有用的信息.此时需要重建误差(reconstrcuterror)作为模型的损失:

$$\mathcal{L}_{reconstruct}=\|Decoder(Encoder(X))-X\|_2 \tag{13}$$

当向量被 Decoder 解码之后,将解码后的向量送入一个新的 Encoder 中.这个 Encoder 需要和之前的 Encoder 的结构保持一致.在设计损失函数时,加入该 Encoder 和之前 Encoder 得出的表示向量之间的距离来让第 2 个 Encoder 尽可能去拟合第一个 Encoder 得到的结果,这就引入了拟合误差:

$$\mathcal{L}_{fit}=\|Encoder(Decoder(Encoder(X)))-Encoder(X)\|_2 \tag{14}$$

这样,当新的数据到来时,如果是和之前的数据分布相同的数据,第 1 个 Encoder 和第 2 个 Encoder 结果之间的差距会尽可能小;而当异常的数据到来时,因为第 2 个 Encoder 从来没有见过异常的数据,就会使两个 Encoder 之间的误差变大.因此,可以直接使用两个 Encoder 之间的误差作为异常分数:

$$score_{anomaly}=\|Encoder(Decoder(Encoder(X)))-Encoder(X)\|_2 \tag{15}$$

这种做法不需要存储聚类中心或查找距离新数据最近的聚类中心,只需保存模型即可.模型的两个输出之间的差可以直接作为最终的异常分数被使用.

## 3 实验评估

本节在多个数据集上评估并比较本文的方法和其他方法.首先,在第 3.1 节中介绍在评估中使用的数据集以及数据预处理的方法;第 3.2 节中介绍使用的各项指标以及模型的架构参数;第 3.3 节中对实验结果进行说明并与其他一些方法进行对比.

### 3.1 数据集

在实验中使用 IDS2017、Digg 和 Reddit Hyperlink Network 这 3 个数据集进行实验.这 3 个数据集的基本情况见表 1.

**Table 1** Basic information of datasets

**表 1** 数据集基本情况

| 数据集 | 节点数 | 边数 | 时间跨度 | 特征信息 | 是否标注 |
|---|---|---|---|---|---|
| Digg | 30 398 | 87 627 | August 3th 2008~August 6th 2008 | A | 否 |
| IDS2017 | 9 015 | 691 406 | 5/7/2017 8:00~5/7/2017 20:00 | A, X | 是 |
| RedditHyperlink | 55 863 | 858 490 | Jan 2014~April 2017 | A, X | 是 |

3.1.1 IDS 2017

该数据集是加拿大网络安全研究所从多个局域网上的电脑模拟的网络攻击场景中收集到的网络攻击数据[32],它的每一条数据由 75 个特征构成,每一条记录都包含有源 IP、目标 IP 以及时间戳等方便进行动态网络

的建模.为了简化实验设置,本文使用星期三的数据进行实验.星期三的数据中一共有 640 000 条边,包含了 Dos (denial of service)攻击的总计 5 种种类,对设置图流异常检测实验比较有利.在实验过程中,我们将每 5 分钟内的所有边视作一个图.其中,当一个图中存在 200 条以上的攻击边时,认为其是异常图.对于每一个图,我们使用所有的特征作为模型的输入.

#### 3.1.2 Digg

Digg 数据集[33]是由 Digg 社交网站中的用户发帖、回帖信息组成的网络,该数据集中包含 30 360 个节点以及 85 155 条边,其中,最大的节点的度数为 283,节点的平均度数为 5.61,每条边都有其自己的时间戳.值得注意的是,该数据集中不存在标注好的异常,因此我们使用异常注入[5]的方法注入异常数据.在进行子图的划分时,我们使用一定时间段内的边作为图,并在划分完子图后随机选择 10%的子图进行异常注入作为异常图.之后,我们在注入异常的数据集上测试模型.

#### 3.1.3 Reddit Hyperlink

Reddit Hyperlink Network[34]为斯坦福 Snap 实验室整理的大型网络数据集[35]中的一个,该数据集收集了从 2014 年 1 月~2017 年 8 月所有 Reddit 上的不同子话题之间的超链接.一个超链接源于一个子话题并终于另一个子话题.该数据集最早被用来检测不同子话题用户之间发生的争论.我们将其应用于异常检测的设置中并进行处理,将每一天的边的数据看作一个图.数据集中记录有每一条边的情感类型,分为正向情感以及负向情感两种.因为正向情感与负向情感分布较为均匀,在实际的检测中,我们不使用该标记作为区分异常标志;相反,我们直接使用本文的方法在该数据集上进行运行,并作为实例来验证我们的算法可以找出一定的具有实际意义的异常情况.

### 3.2 对比方法

- DeepWalk[9]是经典的图上表示学习算法,其使用随机游走获得节点的上下文,并使用 skip-gram 算法进行节点表示向量的学习;
- Node2vec[36]是 DeepWalk 方法加入搜索 bias 的改进算法;
- SDNE(structural deep network embedding)[37]:该方法使用 Autoencoder 以及结构上的限制来学习高层次的非线性网络结构;
- Spotlight[18]是最新的图流上的异常检测算法,其使用 Hash 函数将图 sketch 到 spotlight 向量空间来扩大异常图和正常图之间的距离,并使用流上的异常检测算法进行异常检测;
- GraphSage[27]:该方法使用聚合、拼接两个步骤结合采样方法实现图信息的提取,可以适用于大网络的状态.对于训练过程,该方法使用 skip-gram 型的损失函数来实现无监督学习;
- NetWalk[19]:该方法使用水库采样来保持图的动态信息,并使用深度 Autoencoder 和 CliqueEmbedding 来进行节点的表示向量的学习.原文中使用 streaming-kmeans 来进行边异常值的打分.

### 3.3 实验结果与分析

#### 3.3.1 准确性

在这一节评估方法的准确性.首先,在 IDS2017 和注入异常的 Digg 数据集上进行测试.其中,对于 IDS2017 使用其周三一天的数据进行网络异常检测.我们将 1 分钟内经过的所有边作为一个子图对数据集进行划分,总共获得了 1 008 个子图.之后,将前一半时间的数据作为训练集训练模型,后一半时间的数据作为测试集来对模型进行测试.对于每一个子图,当图中的被标注为攻击边的数目多于 200 时,认为其为异常图.对于 Digg 数据集,首先将每 100 个时间单位内的边作为那一时刻的图对数据集进行划分,并得出共 124 个子图.之后,将前一半时间的图作为训练集,后一半时间的图作为测试集,并在其上进行模型的训练与测试.在测试集中随机选取 10%的图作为异常图,并在其内注入异常边.异常注入的方法是随机选取图内的 0~3 条边并复制 30 次.之后,在没有异常的训练集上训练模型,并在测试集上测试结果.对于 DeepWalk,Node2vec,SDNE 等表示学习算法,将其在每个图上进行运行并得出边的表示向量,之后使用读取函数从表示向量中读取信息作为整个图的表示;对于

SpotLight,直接将其运行于每个图上并得到图的素描向量;对于 Dynamic-DGI,将其在训练集上进行训练并在测试集上进行测试.对于每个数据集,我们使用最大距离读取函数,并使模型学习 20 轮.对于以上所有方法,使用第 3.3 节中介绍的 3 种异常检测算法对学得的表示向量(或 sketch 向量)进行异常检测,之后计算 AUC(area under curve)值来评估实验结果.对于每个方法,设置表示向量的维度为 512 并运行 10 次取其平均 AUC 值作为实验结果.实验结果见表 2.

**Table 2** Result of accuracy experiment

**表 2** 准确性实验结果

| 对比方法 | IDS 2017 | | | Digg | | |
|---|---|---|---|---|---|---|
| | RRCF | Streaming *k*-mean | EDE | RRCF | Streaming *k*-mean | EDE |
| DeepWalk | 0.66(±0.05) | 0.60(±0.04) | 0.59(±0.06) | 0.65(±0.03) | 0.63(±0.03) | 0.62(±0.04) |
| Node2vec | 0.64(±0.05) | 0.62(±0.05) | 0.61(±0.04) | 0.66(±0.05) | 0.62(±0.06) | 0.65(±0.03) |
| SDNE | 0.66(±0.01) | 0.56(±0.03) | 0.60(±0.03) | 0.76(±0.04) | 0.72(±0.06) | 0.70(±0.03) |
| Spotlight | 0.86(±0.02) | 0.84(±0.03) | 0.77(±0.02) | 0.75(±0.02) | 0.74(±0.02) | 0.69(±0.03) |
| GraphSage | 0.72(±0.04) | 0.73(±0.03) | 0.68(±0.04) | 0.69(±0.03) | 0.67(±0.02) | 0.63(±0.06) |
| NetWalk | 0.78(±0.06) | 0.75(±0.04) | 0.74(±0.03) | 0.73(±0.05) | 0.70(±0.04) | 0.68(±0.05) |
| Dynamic-DGI | **0.91(±0.02)** | 0.86(±0.03) | 0.82(±0.02) | **0.81(±0.01)** | 0.74(±0.02) | 0.73(±0.01) |

由表中数据可以看出:Dynamic-DGI+RRCF 在两个数据集上都取得了最好的效果,其中,在 IDS2017 上比专注于图流的异常检测算法 Spotlight 有 5.8%的提升,在 Digg 上比之有 8%的提升.此外,在其他的两种异常检测算法中,Dynamic-DGI 的表现都比其他方法要好.这说明本文的方法成功抓住了网络的结构以及属性上的异常特征.除此之外,相比较于其他方法,Dynamic-DGI 最终的结果浮动更为稳定.值得注意的是,以随机游走为基础的网络表示学习方法 DeepWalk 以及 Node2vec 在两个数据集上的表现都不好,这说明依靠单纯的经验性指标认为越相近的节点具有越相同的向量表示具有一定偏差,不能够很好地反映节点周围的结构特征.而 SDNE 在 Digg 上的表现较好,这说明它在一定程度上能够抓住节点的邻域结构特征.但是对于 IDS2017,SDNE 的效果比较差,这应该与 IDS2017 的数据集上点的数量相对过少而边的数量相对比较多有关.过多的边使得节点之间的连通度大大增加,使 SDNE 不能够很好地区分节点之间结构的不同.对于 Spotlight,可以看出:除了 Dynamic-DGI 外,该算法表现最好.这是因为该算法专注于解决稠密子图的突然出现(或消失)问题,比较适用于本文两个数据集中的异常的情况.而 Spotlight 仅仅只能抓住结构上的异常,不能对属性上的异常做出处理,因此最终的结果相比于 Dynamic-DGI 要差.对于 GraphSage 来说,虽然其聚合邻域的方法没有使用到随机游走,但是其损失函数的计算方法用到了 skip-gram 模型,从而使学得的向量具有经验性的偏差.相比较而言,Dynamic-DGI 能够同时抓住结构、属性以及变化上的异常,因此最终的效果最好.此外还可以看到,算法在 IDS 2017 上提升的幅度大于 Digg 数据上提升幅度.我们分析认为,这是和数据集本身的状态相关的.IDS 2017 的数据集属于在真实网路数据中采集得到的结果,包含的攻击种类和模式符合正常情况下的攻击情况,对于异常都有比较明确的标记;而在 Digg 数据集中并不包括真实异常的标记,其所标记的异常是使用异常注入算法注入得到的,原来网络中也可能存在有一定数量的异常元素.在进行训练时,模型是在认为只包含正常元素的训练集上进行训练的,因此数据集中本身存在的异常可能会对模型的效果造成潜在的影响.

接下来绘制出图流的异常检测算法 Dynamic-DGI 和 Spotlight 在 IDS 2017 数据集上随时间变化的的异常检测分数以及标注好的结果,来比较算法之间的动态异常检测能力,如图 5、图 6 所示.

由图 5 可以看出,在异常最为集中的第 300~400 的图中,Dynamic-DGI 对异常图的得分普遍很高,比较符合于真实的结果;而对于 200~300 之间有一处突起,这应该是源于在划分图时不同图中数据分布不均匀.虽然该图包含不超过阈值的攻击边,但是因为其本身的网络的变化与其他图不同而造成异常分数较高.而对于 Spotlight 来说,由图 6 可以看到,其在攻击图的范围内的得分普遍与正常图流上的得分相差不大,只是在接近第 400 个图的时候有一处明显的突起,并且连带周围的图的分数也升高.这可能是因为 Spotlight 比较适合于稠密子图比较多时的情况,当稠密子图在短时间内激增但没有达到一定数量时,该算法不能将其很好地区分出来.这也是为什么在 300~400 图之间靠前位置虽然有很多异常图,但是 Spotlight 却没有给这些图很高的分数.即:靠前的这些异

常不是很连续,没有达到 Spotlight 的敏感程度可以接收的范围.

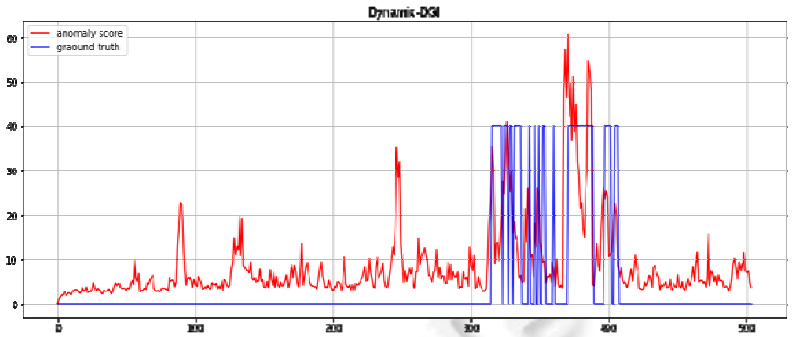



Fig.5 Anomaly score abtained by Dynamic-DGI

图 5 Dynamic-DGI 的异常检测分数

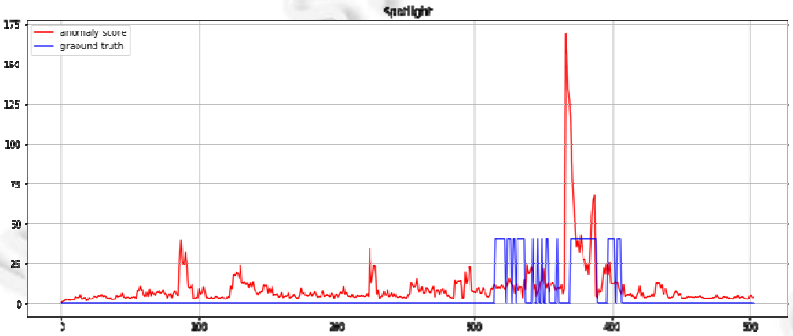



Fig.6 Anomaly score abtained by Spotlight

图 6 Spotlight 的异常检测分数

### 3.4 读取函数比较

本节对读取函数进行比较,读取函数对于获得全图的总结向量具有比较大的影响.实验中主要测试了 3 种读取函数,分别为平均读取函数、最大读取函数和本文提出的最大距离读取函数.这 3 种函数分别对信息进行不同程度的过滤和组合,从而得到表示整个图的表示向量.将 3 种方法应用于 IDS2017 并使用 RRCF 进行异常检测,获得的结果见表 3.

**Table 3** Comparsion of readout functions

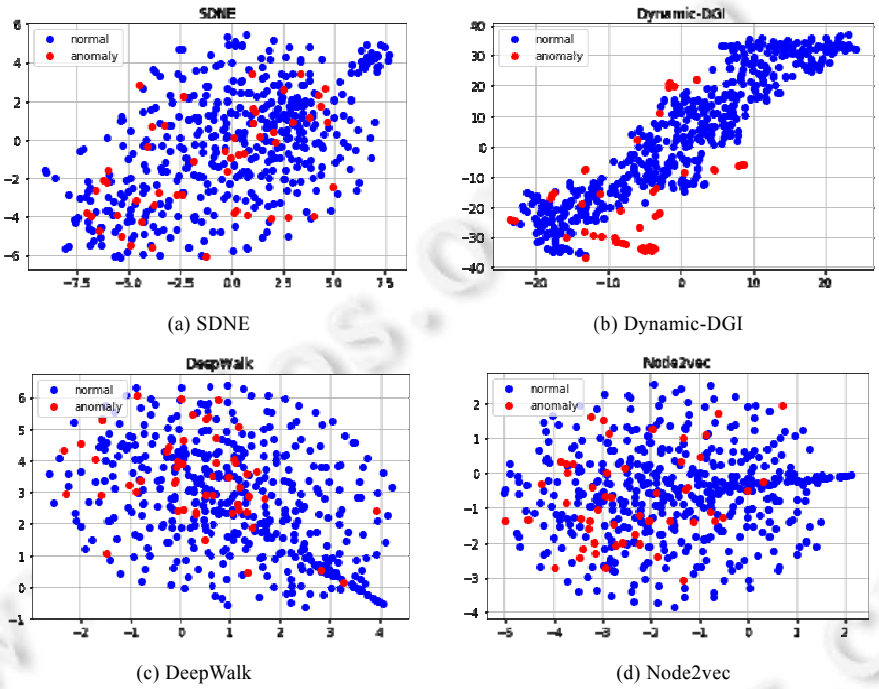
**表 3** 不同读取函数比较

| 读取方法 | Mean | Max | Max-Distance |
|---|---|---|---|
| 结果 | 0.86(±0.02) | 0.90(±0.03) | 0.91(+0.02) |

从表 3 中可以看出,最大距离读取函数获得了最好的结果,对比 Max 读取函数有部分提升,并具有更好的稳定性.而 Mean 读取函数获得了最差的结果.这是因为使用 Mean 的读取函数会像对图像进行平均池化处理时一样损失最显著的特征,从而影响图像分类的精确度;而对于异常检测而言,需要将图的本身最显著的特征尽可能放大,从而可以提取出图最异常的特性.对比于 Max,最大距离读取函数读取的不是图本身的最大化特征而是相比于之前网络的最大化的特征,这能够将不同于之前网络特性的特征挖掘出来,从而可以挖掘出当前图中和之前的图最不同的特性.

### 3.5 聚类效果

本节中展示本文方法得到的隐含表示的聚类效果.好的聚类效果能够在隐空间中将正常图和异常图区分开,从而达到能够区分正常图和异常图的目的.我们分别使用 t-SNE[38]来对 DeepWalk,SDNE,SpotLight 和 Dynamic-DGI 在 IDS 2017 上得到的表示向量进行降维,并将降维的结果显示出来(如图 7 所示).




(a) SDNE (b) Dynamic-DGI

(c) DeepWalk (d) Node2vec

Fig.7 The t-SNE result of the four representation learning methods

图 7 4 种表示学习方法 t-SNE 效果

从图 7 中可以看出,4 种方法对图的表示向量都有聚类的效果.其中,DeepWalk 的结果和 Node2vec 的结果相近,但是二者都不能很好地将异常点和正常点区分开来.可能的原因在第 3.3.1 节有部分说明,即:网络攻击数据集内部边的数量与节点数量的比值相对于普通的数据集来说要多很多,因此节点之间的连通程度较高,使用纯结构的算法不能很好地区分节点.而对于 SDNE,可以看出有了初步的聚类结果,但是不能够将异常图和正常图区分开,因为其聚合邻居的方式没有用到图卷积神经网络,因此获取结构、属性信息的方式存在一定的差距.对于 Dynamic-DGI,可以看出,其将大部分的异常图节点(红色)与正常图节点(蓝色)区分开来,并且所有数据点呈现出多个聚类中心.这说明了与前 3 种方法相比,Dynamic-DGI 对于这种动态的网络形式具有更好的学习和区分能力.

### 3.6 案例分析

在这一节中使用本文的算法在 Reddit Hyperlink 上运行,并说明算法能够发现一些有趣的异常现象.具体的得分情况如图 8 所示.

由图片可以看出,在 100~200,200~300,300~400,400~500 区间内都有图出现非常高的异常值.我们取异常值最高的第 279 张图来查看获取到的异常信息.从图中可以发现,这些异常出现的时间点对应于社团之间负面情绪沟通相对来说比较多的时候.在这些时间里,Reddit 的不同子话题下的用户之间沟通较为频繁,同时,这些沟通之中包含的具有负面情绪的回应也相对较多.而在普通的情况下,大多数的用户沟通都发生在同话题下,社团之间的沟通不多,这就导致了网络结构以及边的属性的差异,因此这些点的异常分数较高.

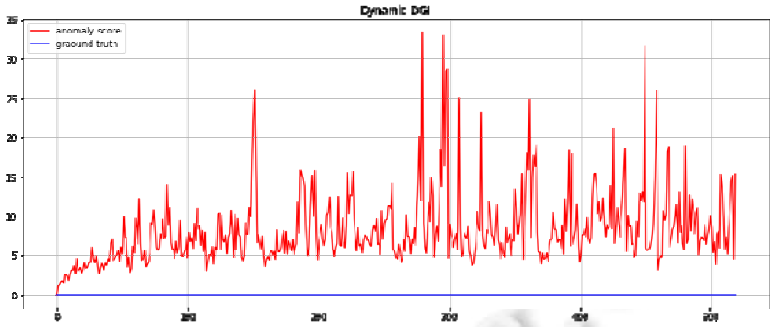



Fig.8 Anomaly score of Reddit Hyperlink network

图 8 Reddit Hyperlink 异常分数

## 4 总结与展望

动态网络异常检测问题具有比较重要的研究地位和实践意义.本文主要研究了动态网络中的异常检测问题,比如发掘正常网络流量中的网络攻击、网络变化中的异常现象等.现存的动态网络异常检测算法主要关注于动态网络的结构特征,这些做法通过比较不同时刻网络结构之间的不同来给出图的异常分数.为了同时兼顾结构信息和网络节点、边的属性信息,本文使用图神经网络来对图数据进行建模,从而能够使算法同时提取网络的结构特征和属性特征,能够挖掘出更多的异常情况.同时,本文引入长短路记忆网络来对网络的变化这一特征进行建模,从而考虑网络变化上的异常.从结果来看,本文的算法相比最好的图流异常检测算法有 5.8%~8%的提升.未来的工作包括优化 Dynamic-DGI 的运行效率,构造特殊的采样方法以使模型适用于更大的网络情况,并且改进模型更新方法使 Dynamic-DGI 适用于在线学习(online learning)的情景,让模型能够对流上的数据进行更好的学习和预测.

**References**:

[1] Carley KM. Dynamic Network Analysis. CMU, 2003.

[2] Eswaran D, Faloutsos C. Sedanspot: Detecting anomalies in edge streams. In: Proc. of the Int'l Conf. on Data Mining (ICDM). 2018. 953−958.

[3] Gupta M, Gao J, Sun Y, *et al*. Integrating community matching and outlier detection for mining evolutionary community outliers. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2012. 859−867.

[4] Heard NA, Weston DJ, Platanioti K, *et al*. Bayesian anomaly detection methods for social networks. The Annals of Applied Statistics, 2010,4(2):645−662.

[5] Noble CC, Cook DJ. Graph-based anomaly detection. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2013. 631−636.

[6] Ranshous S, Shen S, Koutra D, *et al*. Anomaly detection in dynamic networks: A survey. Wiley Interdisciplinary Reviews: Computational Statistics, 2015,7(3):223−247.

[7] Savage D, Zhang X, Yu X, *et al*. Anomaly detection in online social networks. Social Networks, 2014,39:62−70.

[8] Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: A survey. Data Mining and Knowledge Discovery, 2015,29(3):626−688.

[9] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2014. 701−710.

[10] Guha S, Mishra N, Roy G, *et al*. Robust random cut forest based anomaly detection on streams. In: Proc. of the Int'l Conf. on Machine Learning (ICML). 2016. 2712−2721.

[11] Bronstein MM, Bruna J, Lecun Y, *et al*. Geometric deep learning: Going beyond euclidean data. IEEE Signal Processing Magazine, 2017,34(4):18−42.

[12] Monti F, Boscaini D, Masci J, *et al*. Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2017. 5115−5124.

[13] Zhou J, Cui G, Zhang Z, *et al*. Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434, 2018.

[14] Hochreiter S, Schmidhuber JU. Long short-term memory. Neural Computation, 1997,9(8):1735−1780.

[15] Aggarwal CC, Zhao Y, Philip SY. Outlier detection in graph streams. In: Proc. of the 27th Int'l Conf. on Data Engineering (ICDE). 2011. 399−409.

[16] Ranshous S, Harenberg S, Sharma K, *et al*. A scalable approach for outlier detection in edge streams using sketch-based approximations. In: Proc. of the 2016 SIAM Int'l Conf. on Data Mining (SDM). 2016. 189−197.

[17] Manzoor E, Milajerdi SM, Akoglu L. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2016. 1035−1044.

[18] Eswaran D, Faloutsos C, Guha S, *et al*. Spotlight: Detecting anomalies in streaming graphs. In: Proc. of the 24th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2018. 1378−1386.

[19] Yu W, Cheng W, Aggarwal CC, *et al*. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In: Proc. of the 24th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2018. 2672−2681.

[20] Huyan K, Fan X, Yu LT, Luo ZX. Graph based neural network regression strategy for facial image superresolution. Ruan Jian Xue Bao/Journal of Software, 2018,29(4):914−925 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5405.htm [doi: 10.13328/j.cnki.jos.005405]

[21] Qu Q, Yu HT, Huang RY. Graph convolutional network based social network Spammer detection technology. Chinese Journal of Network and Information Security, 2018,30(5):43−50 (in Chinese with English abstract).

[22] Ning SQ, Guo MZ, Ren SJ. A semi-supervised method for cancer clinical outcome prediction based on graph convolution network. Intelligent Computer and Applications, 2018,8(6):44−48 (in Chinese with English abstract).

[23] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). 2016. 3844−3852.

[24] Henaff M, Bruna J, Lecun Y. Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163, 2015.

[25] Kipf TN, Welling M. Semi-Supervised classification with graph convolutional networks. In: Proc. of the Int'l Conf. on Learning Representations (ICLR). 2017.

[26] Veličković P, Cucurull G, Casanova A, *et al*. Graph attention networks. In: Proc. of the Int'l Conf. on Learning Representations (ICLR). 2018.

[27] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). 2017. 1024−1034.

[28] Hjelm RD, Fedorov A, Lavoie-Marchildon S, *et al*. Learning deep representations by mutual information estimation and maximization. In: Proc. of the Int'l Conf. on Learning Representations (ICLR). 2018.

[29] Veličković P, Fedus W, Hamilton WL, *et al*. Deep graph infomax. In: Proc. of the Int'l Conf. on Learning Representations (ICLR). 2018.

[30] Weber M, Domeniconi G, Chen J, *et al*. Anti-Money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. In: Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD), Workshop on Anomaly Detection in Finance. 2019.

[31] Xu K, Li C, Tian Y, *et al*. Representation learning on graphs with jumping knowledge networks. In: Proc. of the Int'l Conf. on Machine Learning (ICML). 2018.

[32] Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proc. of the 4th Int'l Conf. on Information Systems Security and Privacy (ICISSP). 2018.

[33] Akoglu L, Faloutsos C. Anomaly, event, and fraud detection in large network datasets. In: Proc. of the 6th ACM Int'l Conf. on Web Search and Data Mining (WSDM). 2013. 773−774.

[34] Kumar S, Hamilton WL, Leskovec J, *et al*. Community interaction and conflict on the Web. In: Proc. of the 2018 World Wide Web Conf. on World Wide Web (WWW). 2018. 933−943.

[35] SNAP Datasets: Stanford Large Network Dataset Collection. Stanford University, 2014.

[36] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2016. 855−864.

[37] Wang D, Cui P, Zhu W. Structural deep network embedding. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). 2016. 1225−1234.

[38] Laurens VDM, Geoffrey H. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008,9(11):2579−2605.

**附中文参考文献**:

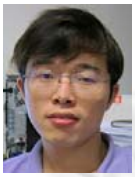
[20] 呼延康,樊鑫,余乐天,罗钟铉.图神经网络回归的人脸超分辨率重建.软件学报,2018,29(4):914−925. http://www.jos.org.cn/1000-9825/5405.htm [doi: 10.13328/j.cnki.jos.005405]

[21] 曲强,于洪涛,黄瑞阳.基于图卷积网络的社交网络 Spammer 检测技术.网络与信息安全学报,2018,30(5):43−50.

[22] 宁世琦,郭茂祖,任世军.基于图卷积网络的癌症临床结果预测的半监督学习方法.智能计算机与应用,2018,8(6):44−48.

**郭嘉琰**(1997－),男,河南南阳人,博士生,CCF 学生会员,主要研究领域为图数据挖掘,图机器学习,图数据管理.

**张岩**(1970－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为信息检索,文本挖掘,大数据分析,网络科学.



**李荣华**(1985－),男,博士,副教授,博士生导师,CCF 专业会员,主要研究领域为图数据管理,图数据挖掘,社交网络分析,图机器学习,图计算系统.

**王国仁**(1966－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为不确定数据管理,数据密集型计算,可视媒体数据管理与分析,非结构化数据管理,分布式查询处理与优化技术(主要包括传感器网络和 P2P 对等计算),生物信息学.