

基于知识图谱和污点传播的网络攻击检测方法

黄明义¹, 邹福泰¹, 周纸墨¹, 张亮²

(1. 上海交通大学网络空间安全学院, 上海 200240; 2. 国家电网有限公司华东分部, 上海 200120)

摘要: 随着计算机和网络通信技术的迅猛发展, 大数据背景下的网络攻击检测日益受到关注。尽管机器学习技术在此领域取得了良好效果, 但数据集的标注和训练问题难以得到解决。传统的信念传播算法 (belief propagation algorithm) 虽然被广泛应用于图攻击检测, 但其缺乏对节点和边类型的区分, 且在处理恶意节点远少于良性节点场景时表现一般。为解决这些问题, 提出了一种基于知识图谱和污点传播的网络攻击检测方法 CDT (community detection and taint propagation), 通过定义 IP 地址、域名和文件 3 种实体, 建立实体间的直接与间接关系来构建知识图谱, 并在半监督环境下应用 Louvain 社区发现算法划分知识图谱, 提取恶意实体相关的子图。另外, 还提出了一种新型的污点传播算法, 基于实体间的关系来推算节点的恶意值, 从而能有效地发现恶意和受害实体, 并可视化攻击路径。实验结果表明, 无论是在仿真实验环境还是权威数据集上, CDT 都表现出了卓越的性能, 远优于传统的信念传播算法。特别是在恶意节点数量较少的情况下, CDT 能够有效地检测攻击行为, 且精度和召回率均远高于传统方法。这不仅证明了 CDT 在网络攻击检测中的卓越性能, 还表明它能够在复杂的网络环境中有效识别恶意行为, 展现出其在实际应用中的优越性。

关键词: 网络攻击检测; 知识图谱; 社区发现; 污点传播

中图分类号: TP309

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2025011

Network attack detection method based on knowledge graph and taint propagation

HUANG Mingyi¹, ZOU Futai¹, ZHOU Zhimo¹, ZHANG Liang²

1. School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

2. East Branch of State Grid Corporation of China, Shanghai 200120, China

Abstract: With the rapid development of computer and network communication technologies, research on network attack detection in the context of big data has increasingly gained attention. Although machine learning techniques

收稿日期: 2024-04-08; 修回日期: 2024-07-10

通信作者: 邹福泰, zoufutai@sjtu.edu.cn

基金项目: 国家自然科学基金 (61831007); 国家重点研发计划 (2020YFB1807500)

Foundation Items: The National Natural Science Foundation of China (61831007), The National Key R&D Program of China (2020YFB1807500)

引用格式: 黄明义, 邹福泰, 周纸墨, 等. 基于知识图谱和污点传播的网络攻击检测方法[J]. 网络与信息安全学报, 2025, 11(1): 151-164.

Citation Format: HUANG M Y, ZOU F T, ZHOU Z M, et al. Network attack detection method based on knowledge graph and taint propagation[J]. Chinese Journal of Network and Information Security, 2025, 11(1): 151-164.

have achieved promising results in this field, issues related to dataset labeling and training have remained challenging. Traditional belief propagation algorithms, while widely used in graph-based attack detection, lacked the distinction between node and edge types and performed inadequately in scenarios where malicious nodes were far fewer than benign nodes. To address these issues, a network attack detection method based on knowledge graphs and taint propagation, referred to as CDTP (community detection and taint propagation), was proposed. In this method, three types of entities (IP addresses, domain names, and files) were defined to establish both direct and indirect relationships between entities, and a knowledge graph was constructed. In a semi-supervised setting, the Louvain community detection algorithm was utilized to partition the knowledge graph and extract subgraphs related to malicious entities. Additionally, a novel taint propagation algorithm was introduced, which inferred the maliciousness score of nodes based on the relationships between entities, thereby effectively detecting malicious and victim entities and visualizing attack paths. Experimental results demonstrate that CDTP outperforms the traditional belief propagation algorithm in both simulated environments and authoritative datasets, showing superior performance. Particularly in scenarios where the number of malicious nodes is small, CDTP effectively detects attacks with significantly higher precision and recall compared to traditional methods. This proves that CDTP exhibits outstanding performance in network attack detection and effectively identifies malicious behaviors in complex network environments, demonstrating considerable superiority in practical applications.

Keywords: network attack detection, knowledge graph, community detection, taint propagation

0 引言

随着数字通信、物联网和云计算的普及,网络威胁的攻击面呈指数级增长,网络安全的重要性达到了前所未有的高度^[1]。在当前错综复杂的全球网络威胁态势下,各种新型攻击手段不断涌现,攻击方式也愈加复杂,网络攻击检测已成为一个热门且重要的领域。当前,全流量检测是一种有效的网络攻击检测方法,主要采用机器学习算法对流量进行异常行为检测^[2]。然而,全流量检测依赖人工进行特征提取,对未知或变种攻击的识别能力也较弱^[3]。近年来,基于图模型的网络攻击检测方法因能够更有效地捕捉复杂攻击模式和网络结构中的异常关系,有效提高检测精度和效率,逐渐受到研究者的青睐。不过,该方法也面临两大关键问题:①海量数据的处理和标定;②攻击场景的多样性和复杂化。

针对传统基于图模型的网络攻击检测方法在大数据环境下样本难以标定、场景复杂的问题,本文基于半监督环境提出了一种名为 CDTP (community discover and taint propagation) 的基于知识图谱和污点传播的网络攻击检测方法。该方法主要包括以下几个步骤:知识图谱构建、基于知识图谱的社区发现、基于恶意知识图谱子图的污点传播,以及最终的网络攻击检测。关于知

识图谱的构建,本文创造性地引入了 IP 地址、文件和域名 3 种实体种类,并且考虑了实体间潜在的直接和间接关系。在构建完知识图谱之后,本文基于 Louvain 社区发现算法,划分恶意知识图谱子图。最后,本文使用污点传播机制,检测恶意子图内隐藏的恶意实体和受害实体。

本文的主要贡献如下。

1) 将知识图谱应用于网络攻击检测,考虑日志中实体间的直接关系以及域名与文件间的间接关系,深度挖掘网络流量日志中的有用信息。

2) 应用 Louvain 社区发现算法对知识图谱进行社区划分,将庞大的知识图谱划分为多个恶意知识图谱子图,并采用一种新型的污点传播算法来发现恶意实体,在此过程中无须对全部测试数据进行标定,避免了恶意节点标注的难题,适用于真实复杂的网络环境。

3) 在搭建的仿真环境数据集和权威数据集上进行实验验证,实验结果表明,本文提出的方法相比传统的信念传播算法具有更高的准确率,说明其在网络攻击检测方面具有优越性。

1 相关工作

本节介绍近年来基于图模型技术的网络攻击检测方法,包括传统图模型技术和知识图谱技术在识别复杂攻击模式和提高检测精度方面的应用

进展。

基于图模型的网络攻击检测方法将网络中的各种实体及其关系表示为图结构,通过分析图结构和图中的模式,检测异常行为或攻击。这种方法能够捕捉到网络中复杂的关系和攻击模式,适用于发现隐藏在正常流量中的异常行为。

传统图模型通过分析网络节点和连接的结构特征来识别异常行为和潜在攻击。Chau等^[4]提出了一种检测恶意软件的新技术Polonium,其使用基于可扩展信念传播算法的大规模图推理技术,通过推断文件的声誉,将低声誉文件标记为恶意文件。Sun等^[5]提出基于概率的零日攻击路径识别方法,首先构建对象实例图作为超图,并基于对象实例图构建贝叶斯网络,使用贝叶斯网络计算对象实例被感染的概率,将高概率实例连接起来生成攻击路径。Lyu等^[6]提出了一种基于溯源数据的智能网络攻击检测方法,使用异质图建模和语义向量表示,从大量系统事件中高效检测网络攻击。其在实际数据集上的表现优于现有学习型和基于规则的检测模型。Sosnowski等^[7]提出了一种基于图模型的TLS生态系统分析方法,结合概率威胁传播算法,利用服务器、域名和证书之间的关系来有效识别潜在的恶意实体,从而提升对未知威胁的检测能力。基于传统图模型的网络攻击检测方法尽管在特定场景下取得了不错的效果,但在处理异构数据时,模型的泛化能力有限,在大规模数据集上检测的效率和精度也不高。

近年来,越来越多的研究者采用知识图谱技术检测网络攻击。知识图谱是一种语义链接的实体,可以将人们对物理世界的认知转化为语义信息,进而使计算机能够以结构化的方式理解这些信息^[8]。知识图谱能更直观地表示网络攻击中的复杂关联关系,更容易发现潜在威胁及异常模式,相比传统图模型具有更高的灵活性和准确性。Ismail等^[9]通过对19种传统机器学习技术进行深入评估,证明集成知识图谱能显著提升Web应用攻击检测的准确性和分析性能。

Milajerdi等^[10]提出了基于系统日志的知识图谱方法,设计了高级持续性威胁(advanced and persistent threat, APT)检测系统HOLMES,通过

检测不同APT阶段的系统行为恶意性来判断APT活动,但该方法仅依据已知APT攻击进行阈值划分,未能有效检测新型攻击。Najafi等^[11]提出基于开源和网络威胁情报的半监督知识图谱方法Malrank,以及利用异构信息网络对主机级系统事件进行建模的方法MalLink^[12],使用图推理算法评估实体的恶意性,但其在日志收集和算法初始权重设计上存在局限性。Ren等^[13]提出了一种基于图卷积神经网络的APT攻击检测模型,通过构建知识图谱和处理图特征,提高了检测准确率,从而提升了APT攻击检测的效果。Qi等^[14]引入了网络安全知识图谱用于复合攻击检测,提出了动态聚类机制的多维数据关联分析算法和基于最优到达路径查询的攻击链补全-剪枝方法,从而提高了复合攻击分析的效率和准确性。

如今,网络攻击的隐蔽性越来越好,常常混杂在海量流量中,难以确切标定。同时,网络攻击技术在不断发展,对新型攻击的识别愈加重要。上述方法要么受限于海量数据集的标定,要么无法对新型攻击进行检测,难以应对越来越复杂的网络安全挑战。

本文将知识图谱技术应用于网络攻击检测,提出一种名为CDTP的检测方法。该方法使用Louvain社区发现算法划分恶意知识图谱子图,结合污点传播模块发现恶意实体,完成网络攻击检测。在整个过程中,无须对所有测试数据进行标定,减少了数据标定的困难。另外,本文方法有较强的适用性,不仅可以满足当前网络攻击复杂多样的检测需求,还能够识别新的恶意节点。

2 CDTP的工作流程与技术实现

本节将介绍CDTP的工作流程以及各个处理模块的具体实现方式。

2.1 CDTP的系统架构

CDTP的系统架构如图1所示,包含以下6个部分。

1) 流量预处理: 基于Zeek框架对流量进行预处理,得到日志文件,选取其中的部分日志文件以供后续处理。

2) 知识图谱构建: 使用Spark计算引擎对得

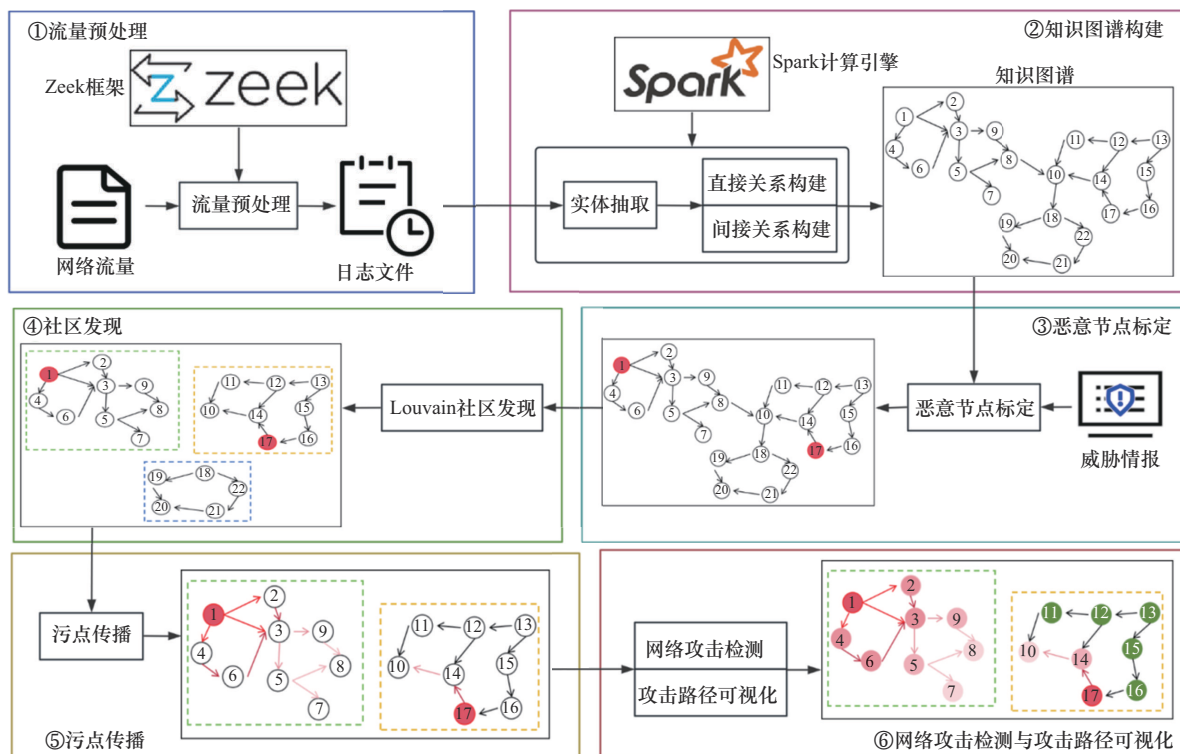


图1 CDTP的系统架构

Figure 1 System architecture of CDTP

到的日志文件进行分析, 从日志中抽取IP地址、域名和文件3种实体, 分析实体间的直接与间接关系, 完成知识图谱的构建。

3) 恶意节点标定: 结合威胁情报信息, 对知识图谱中的部分已知恶意节点进行标定。

4) 社区发现: 对构建好的知识图谱, 依据实体间的相关性进行社区划分, 根据部分恶意实体标定的结果, 快速定位受恶意实体影响较大的社区, 从原知识图谱中挖掘恶意知识图谱子图。

5) 污点传播: 使用新型的污点传播算法, 结合半监督环境下已经标定的部分恶意实体, 计算出恶意知识图谱子图中所有实体的恶意值。

6) 网络攻击检测与攻击路径可视化: 根据实体的恶意值大小检测网络攻击, 并结合知识图谱中的路径分析完成攻击路径可视化。

2.2 知识图谱构建

2.2.1 实体抽取

本文使用流量分析器Zeek (原名Bro) 框架^[15]的文件还原和日志抽取功能, 对传入的Pcap数据包进行处理, 得到对应的日志文件, 并选取其中的Conn.log (记录实体连接信息)、

Dns.log (记录域名解析信息)、Http.log (记录HTTP请求与响应信息)、Files.log (记录实体文件传输信息) 4个日志文件用于后续处理。

得到日志文件后, 使用Spark计算引擎^[16]按行分析日志信息, 抽取实体。本文共分析IP地址、域名和文件3种实体, 即实体 $V \in [IP, Domain, File]$ 。

如果两个实体间存在某种相关关系, 就在知识图谱中添加一条边用来连接这两个实体。这里引入了边权重变量 $\omega (\omega \in [0, 1])$, 用于表示两个实体之间的连接强度。具体来说, 两个实体之间的连接信息越多, 说明它们之间的相关性越强, ω 的值就越大。

根据日志中实体之间的联系情况^[17], 将知识图谱的关系构建划分为直接关系构建和间接关系构建两大类。在直接关系构建中, 关注直接连接的实体之间的关系; 而在间接关系构建中, 通过中间实体或路径建立实体之间的关系。这种划分有助于更细致地捕捉实体之间的关联性, 使得知识图谱的结构更为丰富。

2.2.2 直接关系的构建

对日志文件按行分析, 可得到所抽取的实体

间的关系。例如，若两个IP间存在TCP连接记录，则在两个IP实体间构建直接关系。

边的类型代表了不同类型的边的相关性。初始阶段，不同类型的边被赋予不同的权重，以表示边之间相关性的强弱。例如，Dns.log中IP与其对应解析的域名Domain间存在强相关关系，而日志记录中域名Domain与请求解析该域名的IP之间只存在弱相关关系。直接关系权重设置如表1所示。

对于强相关的边类型，边的权重直接赋值为1。对于弱相关的边类型，需要继续分析日志信息，修改权重。本文中记录两实体间的连接次数为 c ，每次日志中再次出现两个实体的连接信息时，令 $c=c+1$ ，最后使用改进的tanh激活函数对 c 进行归一化处理得到边权重，如式(1)所示。

$$\omega = \frac{e^{\mu c} - e^{-\mu c}}{e^{\mu c} + e^{-\mu c}} \quad (1)$$

为了减小 c 过大对归一化效果的影响，添加了幂次参数 μ 修正权重数值得到边权重 ω 。当 c 较小时，增大 μ ，以增加 c 对最终权重的处理影响；当 c 较大时，减小 μ ，以缩小 c 对最终权重的处理影响。经过tanh激活函数处理后，权重 ω 映射到 $[0,1]$ 区间。

2.2.3 间接关系的构建

有的实体之间按照日志记录不存在直接交互关系，但它们之间可能存在一定的隐藏相关性。本文通过捕捉域名与文件实体间的隐藏相关性，构建实体间接关系。

(1) 域名间接关系构建

本文采用Jaccard相似度算法进行域名相似

性分析。该算法认为归属同一恶意家族的域名会存在相同的实体访问记录，实体记录的相似度越高，则域名属于同一家族，最后归于同一社区的概率越高。

Jaccard相似度算法的计算式如式(2)所示，首先计算两域名 D_1 、 D_2 间的 $J(D_1, D_2)$ 值，并与设定的阈值 S 进行对比。若 $J(D_1, D_2)$ 大于 S ，则认为 D_1 和 D_2 之间具有间接关系，在实体 D_1 和 D_2 之间引入关联边。

$$D_1 \rightarrow \text{list}_1 \{ \text{IPs} \}, D_2 \rightarrow \text{list}_2 \{ \text{IPs} \},$$

$$\omega = J(D_1, D_2) = \frac{|\text{list}_1 \cap \text{list}_2|}{|\text{list}_1 \cup \text{list}_2|} > S \quad (2)$$

(2) 文件间接关系构建

考虑到文件相似性分析常用的SimHash只能从文本上对文件相似性进行判断，本文采用IDA+BinDiff工具^[18]从函数角度计算文件间的相似性得分，判断是否存在间接关系。

使用IDA反汇编工具对文件进行处理，可以导出BinExport格式文件，同时通过调用BinDiff插件，导出反汇编结果。BinDiff是IDA的一款插件，通过比较二进制文件的函数和基本块，计算文件间的相似性得分。如果使用BinDiff计算出的相似性得分大于设定的阈值，那么认为两个文件之间存在间接关系。

2.3 社区发现

2.3.1 恶意节点标定

在本节中，利用已知的恶意实体数据集对知识图谱中实体的恶意值进行标定，从而为后续社区发现算法的执行提供半监督环境。

本文建立的实体黑名单涵盖了自己收集的恶

表1 直接关系权重设置
Table 1 Direct relationship weight setting

日志名称	实体连接类型	边类型	关系强度	初始权重
Conn.log	IP-IP (建立连接的IP)	ConnIP-IP	弱	0.1
Http.log	IP-IP (HTTP交互的IP)	HttpIP-IP	弱	0.1
Dns.log	ScrIP-Domain (IP请求解析Domain)	ScrIP-domain	弱	0.1
Files.log	File-recvIP (发送的文件与接收文件的IP)	File-recvIP	弱	0.1
Files.log	sendIP-recvIP (发送与接收文件的IP)	sendIP-recvIP	弱	0.1
Dns.log	Domain-resIP (Domain与解析的IP)	Domain-resIP	强	1
Dns.log	Domain-cname (domain与对应的cname)	Domain-cname	强	1
Files.log	SendIP-file (发送文件的IP与文件)	SendIP-file	强	1

意实体数据集以及实验评估时确定为恶意的实体(如发动攻击的实体等),共有 800 余万条攻击标记,包含了恶意的 IP 地址和域名、恶意文件哈希值。本文利用该数据集来标定图中实体的恶意值。此外,本文还使用了 Alexa 的 Top10 000 域名作为白名单,在构图时预先过滤掉这些实体,从而减小图的大小。

2.3.2 Louvain 算法原理

社区发现算法要求将图划分为多个社区,其中属于同一社区的实体密集连接,属于不同社区的实体稀疏连接。目前主流的社区发现算法有 GN、Louvain、图卷积等^[19]。Louvain 算法是一种经典的社区发现算法^[20],算法收敛快,适合大量节点关联分析,可用于构建含有边权重的社区场景。本文采用 Louvain 算法进行知识图谱的社区划分。

Louvain 算法产生的分区的质量通常由分区的模块度 $Q(Q \in [-1, 1])$ 来衡量,代表了社区内部节点之间相比于各社区之间连接的紧密程度。设 $G=(V, E)$ 是构建的知识图谱,其中 V 是节点集合, E 是边集合。设 ω 是知识图谱网络的邻接矩阵, ω_{ij} 表示节点 i 与节点 j 之间的连接权重。设 k_i 为实体 i 与其邻居之间相连所有边的权重之和,即

$$k_i = \sum_{ij} \omega_{ij} \quad (3)$$

m 表示知识图谱中所有边的权重之和,计算方式如下。

$$m = \frac{1}{2} \sum_{ij} \omega_{ij} \quad (4)$$

定义社区 c_i 为节点 i 所在的社区, $\delta(c_i, c_j)$ 为示性函数,即如果节点 i 和节点 j 属于同一个社区,那么 $\delta(c_i, c_j)$ 为 1, 否则为 0。模块度 Q 被定义为

$$Q = \frac{1}{2m} \sum_{ij} \left[\omega_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \quad (5)$$

其中, $\frac{k_i k_j}{2m}$ 是期望连接权重,模块度 Q 就是实际连接权重与期望连接权重之差的总和除以 $2m$ 。

为了衡量将某一孤立实体 i 加入某一分区 c 中的合理性,往往使用模块度增益 ΔQ 来表示,它的定义如下。

$$\Delta Q = \left[\frac{\sum i \cdot n + 2k_{i, in}}{2m} - \left(\frac{\sum \text{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum i \cdot n}{2m} - \left(\frac{\sum \text{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (6)$$

其中 $\sum i \cdot n$ 表示分区 c 内部所有边的权重之和, $\sum \text{tot}$ 表示分区 c 内所有节点的度数之和, m 表示所有边的权重累加之和, $k_{i, in}$ 为分区 c 中与实体 i 相连的所有边的权重和。式(6)等号右边的前半部分为把实体 i 加入分区 c 后的模块度,后半部分为实体 i 独立于分区 c 的模块度,模块度增益 ΔQ 表示这两者的差值。如果 $\Delta Q > 0$, 表示把实体 i 加入分区 c 后模块度增加,需要将实体 i 加入分区 c ; 反之,需要将实体 i 与分区 c 分离。

2.3.3 Louvain 算法流程

以一个有 N 个实体的加权图为例, Louvain 算法的执行过程被分为以下两个步骤。

步骤 1: 首先,为图中的每个实体分配一个不同的社区,即初始状态一共有 N 个社区。然后,对于每个实体 i ,考虑它的邻居实体 j ,尝试将 i 加入 j 所在的社区,计算加入前后的模块度增益 ΔQ ,并记录使得模块度增益 ΔQ 最大的邻居实体 $j_{\max}$ 。如果 $\Delta Q_{\max} > 0$, 那么将实体 i 加入 $j_{\max}$ 所在的社区; 否则,保持实体 i 所在的位置不变。重复以上操作,直到每个实体所属的社区不再改变。

步骤 2: 将步骤 1 中形成的各个社区折叠成一个大的超节点,每个社区作为一个新的节点出现。对于新构建的网络,新节点的自我权重是原先同一社区内的节点之间所有边的权重之和,连接不同社区的边的权重则转化为连接新节点之间的边的权重。

重复以上两个步骤,直到整个知识图谱的模块度不再发生变化,最终社区数量趋于稳定。Louvain 算法实现过程示意如图 2 所示。

使用 Louvain 算法可以将整个知识图谱划分为多个社区。如果一个社区中存在恶意实体,那么将该社区包含的实体及关系从整个知识图谱中抽取出来,得到恶意知识图谱子图,用于后续的研究。

2.4 污点传播

本节基于第 2.3 节构建的恶意知识图谱子图,

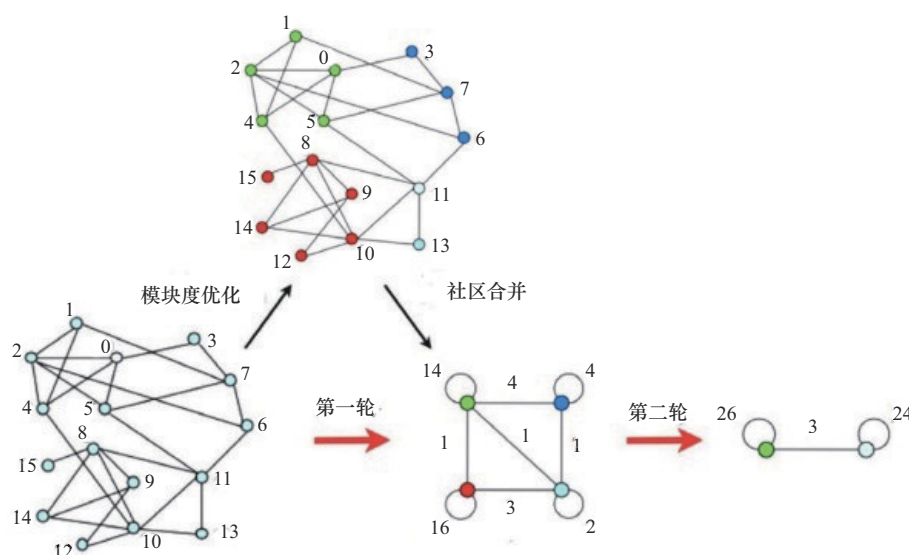


图2 Louvain算法实现过程示意

Figure 2 Louvain algorithm implementation process schematic

采用污点传播算法完成恶意子图内的隐藏恶意实体和受害实体的发现。

污点传播起源于Perl的Tainting机制^[21]，通过将外部不可信数据标记为污点来分析潜在的恶意实体或受害实体，是发现系统潜在脆弱性的重要方法。

信念传播算法（belief propagation algorithm）^[22]是一种广为流传的图推理算法，也被广泛应用于污点传播。信念传播算法通过在图中各节点之间迭代传递恶意值，根据每个节点和其邻近节点的先验知识，推算出该节点的恶意值。然而，信念传播算法并不区分节点和边的类型，且希望构建的图相对平衡，并不适用于本文构建的知识图谱。除信念传播算法外，其他的污点传播算法要么无法在边上引入权重，要么无法定义节点和边的类型，同样不适用于本文构建的知识图谱。

为解决这些问题，CDTP参考并改进了文献^[11]中在半监督环境下使用的静态污点传播算法。该算法基于有向图，根据实体之间的相互关联性，采用边权值和恶意值迭代，推算出实体最终的恶意值。相比传统的污点传播算法，本文提出的污点传播算法有以下优点。

1) 考虑了不同节点和边的类型对污点传播的影响，解决了信念传播算法无法处理节点和边类型的局限性，更加适用于本文构建的知识图谱。

2) 能够在边上引入权重，使得传播过程能够更准确地反映实体之间的关系，提高了算法的灵活性和适应性。

3) 在半监督环境下进行了优化，增强了算法在动态和复杂数据环境中的表现能力。

CDTP算法的具体流程如下。

假设 $s(x)$ 为实体 x 的恶意值，定义 $s(x)$ 为

$$s(x) = c(x) \cdot s^0(x) + (1 - c(x)) \frac{\sum_{y \in N(x)} \sum_{t \in T_{yx}} (\hat{w}_{yx}(t) \cdot s(y))}{\sum_{y \in N(x)} \sum_{t \in T_{yx}} \hat{w}_{yx}(t)} \quad (7)$$

其中， $s^0(x) \in [0, 1]$ ，表示实体 x 的初始恶意值。若 x 为已知恶意实体，令 $s^0(x)=1$ ，否则 $s^0(x)=0.5$ 。 $c(x)$ 表示 $s^0(x)$ 的初始可信度。如果实体已确定是恶意或是正常的，那么 $c(x)=1$ ；否则 $c(x)=0.5$ 。 $N(x)$ 表示实体 x 的邻居实体构成的集合， T_{yx} 是实体 y 对实体 x 边的类型的集合， $\hat{w}_{yx}(t)$ 表示实体 y 对实体 x 的类型为 t 的边的权重。在本文中，边的类型包括直接关系、域名间接关系和文件间接关系3种。

CDTP中的污点传播模块通过迭代，进行边权值、实体恶意值及其他中间变量的循环迭代计算，得到最终的恶意值。边权值根据原始边权值和第 i 次迭代时节点的恶意评分的加权平均值计算得到，循环迭代的计算式如下。

$$\hat{\omega}_{yx}^{i+1}(t) = \begin{cases} 0, & \hat{\omega}_{yx}^0(t) = 0 \\ k \cdot s^i(y) + (1-k) \cdot \hat{\omega}_{yx}^0(t), & \text{其他} \end{cases} \quad (8)$$

其中, 因子 k 表示调整迭代时, 边权重 $\hat{\omega}_{yx}(t)$ 受初始相关权重 $\hat{\omega}_{yx}^0(t)$ 影响的大小程度, $s^i(y)$ 表示在第 i 次迭代中实体 y 的恶意值。

为了表示在每次迭代中恶意值传播的情况, 引入了恶意值传播量 m_{yx} , 理想情况下它的迭代计算式如下。

$$m_{yx}^{i+1}(t) = s^i(y) \cdot \hat{\omega}_{yx}^{i+1}(t) \quad (9)$$

其中, $m_{yx}^{i+1}(t)$ 表示在第 $i+1$ 次迭代中由实体 y 向实体 x 传播的类型为 t 的恶意值, $\hat{\omega}_{yx}^{i+1}(t)$ 表示第 $i+1$ 次迭代中由实体 y 指向实体 x 的类型为 t 的边权重。

然而, 在有向图的实际污点传播迭代过程中, 假设在第 i 次迭代时, 实体 x 向实体 y 传播了恶意值 $m_{xy}^i(t)$, 那么在第 $i+1$ 次迭代时, 实体 y 会向实体 x 回传 $m_{xy}^i(t)$ 的一部分, 造成“回声”现象。因此, 需要对恶意值传播量的计算式进行修正。在第 i 次传播时, 实体 x 向 y 传播恶意值导致 y 的恶意值变化 $s_x^i(y)$ 为

$$s_x^i(y) = (1 - c(y)) \frac{\sum_{t \in T_{xy}} m_{xy}^i(t)}{\sum_{z \in N(y)} \sum_{t \in T_{zy}} \hat{\omega}_{zy}^i(t)} \quad (10)$$

对恶意值传播量的公式进行修正, 以消除回声。修正后的计算式如下。

$$m_{yx}^{i+1}(t) = [s^i(y) - s_x^i(y)] \cdot \hat{\omega}_{yx}^{i+1}(t) \quad (11)$$

最终第 $i+1$ 次迭代得到的实体 x 恶意值为

$$s^{i+1}(x) = c(x) \cdot s^0(x) + (1 - c(x)) \frac{\sum_{y \in N(x)} \sum_{t \in T_{yx}} m_{yx}^i(t)}{\sum_{y \in N(x)} \sum_{t \in T_{yx}} \hat{\omega}_{yx}^i(t)} \quad (12)$$

综上, CDTF 中的污点传播模块的迭代流程主要包括以下 4 个步骤。

步骤 1: 初始化恶意值数组、权重矩阵和恶意值传播矩阵。

步骤 2: 根据当前的恶意值数组、权重矩阵和恶意值传播矩阵计算出新的恶意值数组和权重矩阵。

步骤 3: 根据上一次迭代的恶意值数组、权重矩阵、恶意值传播矩阵以及新的恶意值数组,

计算出新的恶意值传播矩阵。

步骤 4: 若迭代还未结束, 则返回步骤 2。

在经过 n 次污点传播迭代后, 可以得到所有实体的恶意值。设定一个阈值 T , 如果实体的恶意值大于 T , 那么可以判定该实体为恶意实体。

2.5 网络攻击检测与攻击路径可视化

检测出知识图谱中的所有恶意实体后, 可以根据恶意实体间的关系检测网络攻击和标注攻击路径。如果两个恶意实体间存在直接或间接关系, 就可以判定这两个实体参与了网络攻击, 它们之间的关系可以标注为攻击路径。

攻击路径可视化是对 Louvain 社区发现算法划分的恶意知识图谱子图而言的。首先, 选取不同的图形标识不同的实体类型; 然后, 根据实体的恶意值大小为实体分配不同的颜色, 恶意值低于阈值 T 的实体标为绿色, 高于阈值 T 的实体标为红色, 恶意值越高对应的红色越深。这样构建的知识图谱子图, 能够清晰地展示出图中的恶意实体、受害实体及它们之间的攻击路径。这对于安全工作人员识别系统中的漏洞和弱点, 制订有效的响应计划具有重要意义。

3 实验设计

本节将介绍本文实验的设计方法, 包括实验数据集的获取方式、所对比的基线方法以及各模块的参数设置。

3.1 实验数据集获取

网络检测领域常用的经典数据集有 KDD-Cup1999、DRAPA1998 和 DRAPA2000 等^[23]。然而, 这些数据集较旧, 不能反映最新网络流量情况, 且攻击实体和受害实体相对单一, 无法很好地与知识图谱检测技术相结合。

为了更好、更全面地衡量本文提出的网络攻击检测方法的应用效果, 研究人员构建了一个攻击流量和正常流量混杂的实验环境。该实验环境包含 100 多个恶意实体, 总实体数超过 10 000 个, 覆盖 DDoS、僵尸网络、Web、端口扫描和 Brute SSH 暴力破解 5 种常见攻击。

除此数据集外, 本研究还使用权威数据集 CIC-IDS2017 和 CSE-CIC-IDS2018 验证 CDTF 的性能^[24]。这两个数据集由加拿大网络安全研究

所构建, 类似于真实的流量数据, 包含了正常活动流量和最新常见的攻击流量。为了更好地展示基于知识图谱进行检测的优越性, 本实验选择了数据集中包含实体最多的僵尸网络攻击数据进行检测。

3.2 基线方法

为了验证 CDTP 方法的先进性, 本文将相同实验数据分别使用信念传播算法和 CDTP 方法进行处理, 并对比其结果。

如表2所示, 信念传播算法使用以下参数^[11]进行对比实验。节点概率表示一个节点为恶意或正常的概率, 初始时设定已知恶意节点为恶意的概率为0.99, 正常节点为恶意的概率为0.5。边概率 $\psi_{ij}(x_i, x_j)$ 表示节点 i 处于 x_i 状态且节点 j 处于 x_j 状态的概率。 ε 代表一个扰动值, 与图中实体之间相连边的相关性强弱呈正相关。

表2 信念传播算法参数设置
Table 2 Belief propagation algorithm parameter configuration

(a) 节点概率设置		
节点类型	预测为恶意的概率	预测为正常的概率
恶意	0.99	0.01
正常	0.5	0.5
(b) 边概率设置		
边状态	$\psi_{ij}(x_i, x_j)$	
x_i =正常且 x_j =正常	0.5+ ε	
x_i =正常且 x_j =恶意	0.5- ε	
x_i =恶意且 x_j =正常	0.5- ε	
x_i =恶意且 x_j =恶意	0.5+ ε	

与 CDTP 一样, 信念传播算法同样通过迭代计算节点为恶意的概率。设定迭代次数为10, 经过10次迭代后, 如果节点的恶意概率超过阈值0.5, 则判定节点为恶意节点。

3.3 参数设置

3.3.1 实体关系参数设置

(1) 直接关系参数设置

对于弱相关的连接关系, 将连接次数 c 用 $\tanh$ 激活函数处理, 激活函数预先添加指数 $\mu=0.03$ 以减小 c 的影响, 计算结果作为边的权重。对于强相关的连接关系, 边的权重直接赋值为1, 不受 c 值的影响。

(2) 域名间接关系参数设置

域名间接关系通过域名相似性比较来分析。首先计算两个域名源IP的数量总和 t , 随后使用 $\tanh$ 激活函数将其归一化, 如式(13)所示。将前一步获得的Jaccard系数乘以归一化的值 t' 作为新的Jaccard值。本实验设定阈值为0.7, 即当新的Jaccard系数大于0.7时, 才在两个域名间添加一条边, 边权重为Jaccard系数。

$$t' = \frac{e^{0.1t} - e^{-0.1t}}{e^{0.1t} + e^{-0.1t}} \quad (13)$$

(3) 文件间接关系参数设置

使用IDA+Bindiff工具比较文件的相似性来分析文件间接关系。预先设定相似性的阈值为0.8, 随后将每个文件与其他文件进行相似性比较, 当相似度大于0.8时, 在两个文件之间添加一条边, 边权重为文件相似度。

3.3.2 污点传播模块参数设置

所有已知恶意实体的恶意值 $s_m^0(x)$ 设为1, 恶意值的可信度 $c_m(x)$ 也设为1; 其余所有未知的实体的初始恶意值 $s_b^0(x)$ 均设为0.5, 恶意值可信度 $c_b(x)$ 也均设为0.5。在边权重的初值设定中, k 是描述恶意值沿边传播过程中边权重的影响, 预先设定初值为0.1。回声消除矩阵是为了减少恶意值传播过程中回声的影响, 其初值为0, 表示初始状态没有回声影响。

设定迭代次数为10, 通过污点传播模块的迭代, 最终获得所有节点的恶意值。划分实体恶意度的阈值 T 设为0.5。当节点恶意值大于0.5时, 判定为恶意节点, 小于0.5时判定为良性节点。

4 实验结果

本节将从单攻击模式实验、多攻击模式实验以及权威数据集实验3个方面对CDTP方法的性能进行实验评估。其中, 单攻击模式实验与多攻击模式实验的数据集来源于仿真环境, 前者通过与信念传播算法对比验证CDTP方法的准确性, 后者通过还原复杂多攻击场景中的攻击路径检测该方法的性能, 权威数据集实验则进一步验证CDTP方法的通用性与鲁棒性。

4.1 单攻击模式实验评估

本节基于第3.1节收集的数据集, 在每个Pcap样本只包括一种攻击的半监督场景下, 使用

CDTP 方法发现隐藏的恶意实体和受害实体, 评估该方法的效果。

将收集的 Pcap 流量送入网络攻击检测模型, 经过数据抽取、社区发现和污点传播模块检测, 不同攻击环境下 CDTP 方法检测结果如表 3 所示。

表 3 不同攻击环境下 CDTP 方法检测结果
Table 3 Detection results of CDTP method in different attack environments

恶意子图 攻击种类	精确率	漏检率	误检率	准确率
DDoS 攻击	90.91%	0.10%	3.07%	99.90%
端口扫描	95.24%	0.05%	2.41%	99.90%
Web 攻击攻击	95.24%	0.08%	2.66%	99.95%
BruteSSH 攻击	90.91%	0.11%	2.73%	99.93%
僵尸网络攻击	95.24%	0.06%	2.43%	99.92%
总计	93.46%	0.08%	2.66%	99.92%

由表 3 可以得出以下结论。

1) CDTP 方法整体的检测效果优秀, 精确率平均达到 93.46%, 且漏检情况极少, 漏检率平均低于 0.1%。

2) CDTP 方法误检率相比漏检率稍高。这是因为对于现实环境来讲, 将恶意节点标记为良性节点(漏检)造成的损失比将良性节点标记为恶意节点(误检)造成的损失高得多, 所以实验中降低了恶意实体的检测阈值, 使误检率在可接受的情况下漏检率尽可能低。

3) 将 CDTP 方法应用在不同攻击种类的恶意子图上效果不同, 对端口扫描攻击、Web 攻击和僵尸网络攻击的恶意实体检测效果较好, DDoS 攻击和 BruteSSH 攻击的检测效果较差。出现这一现象的原因可能是端口扫描、Web 和僵尸网络等攻击的流量数据通常相对集中和密集, 容易形成明显的子图特征。相比之下, DDoS 攻击和 BruteSSH 攻击的流量数据较稀疏和分散, 难以形成明确的子图特征, 使得攻击

检测变得困难。

单攻击模式实验检测结果如表 4 所示。从表 4 可以看出, CDTP 方法的准确率、恶意实体召回率及精确率都高于基于知识图谱的信念传播算法, 检测的效率也更高。

4.2 多攻击模式实验评估

为了进一步检测 CDTP 方法在网络攻击检测方面的性能, 本文在校园网真实场景中完成攻击环境的搭建, 在正常流量背景下模拟攻击的发动。本实验共设计了 3 个攻击场景。攻击场景 1 为 Web 漏洞攻击, 受害主机为装有 xampp5.6+dvwa1.9 环境的内网机器, 其作为靶机受到攻击主机通过 SQL 注入方式发起的 Web 服务攻击。攻击场景 2 和场景 3 为暴力破解 SSH、DDoS 和僵尸网络构成的混合攻击, 用于模拟真实攻击情境中的复杂场景。

经过 Zeek 框架抽取日志后, 得到的日志信息如表 5 所示。本文从中选取包含模拟攻击的流量用于后续研究。

表 5 日志信息
Table 5 Log information

日志	大小/MB	日志条目/条
Conn.log	54	309 049
Dns.log	5.3	9 313
Files.log	110	189 696
Http.log	125	186 985

得到日志后, 以半监督方式标定 3 个场景中的部分攻击源 IP。使用 Louvain 算法进行社区发现后, 发现构建的知识图谱中, 来自 3 个实验场景的实体都准确地归入了 3 个相应的恶意社区。恶意实体社区分布结果如表 6 所示, 可以看出, 实验场景中的实体都正确地与同一场景内已提前标定的恶意实体聚集在同一社区, 这为后续的恶意实体检测提供了支持。

表 4 单攻击模式实验检测结果

Table 4 Experimental results for single attack mode

检测方法	准确率	恶意实体召回率	精确率	检测耗时/min
CDTP 方法	99.92%	99.20%	93.46%	10
信念传播算法	98.83%	99.00%	90.47%	30

表 6 恶意实体社区分布结果
Table 6 Malicious entity communities
distribution results

场景	所有恶意实体	预标定恶意实体	实体划分结果
1	112.*.249.56 10.0.0.209	112.*.249.56	112.*.249.56 10.0.0.209
2	202.*.1.63 202.*.1.154 202.*.1.156 202.*.7.13 10.0.0.200~ 10.0.0.203	202.*.1.63 202.*.1.154	202.*.1.63 202.*.1.154 202.*.1.156 202.*.7.13 10.0.0.200~ 10.0.0.203
3	202.*.1.65 202.*.7.14 10.0.0.205~ 10.0.0.208	202.*.1.65	202.*.1.65 202.*.7.14 10.0.0.205~ 10.0.0.208

(1) 场景 1

经过污点传播模块迭代后，攻击场景 1 中的所有实体恶意值被确定。分析结果发现，除了攻击主机与受害主机外，其余实体的恶意值均小于 0.5。场景 1 恶意值分布如表 7 所示。

表 7 场景 1 恶意值分布

Table 7 Distribution of malicious values in scenario 1

恶意值区域	实体数量	实体分析
{ 1 }	2	攻击主机：112.*.249.56 受害主机：10.0.0.209
[0.4, 0.5)	9	访问次数较多的正常域名或 IP： 47.*.212.165 等
[0.2, 0.4)	85	访问次数较少的正常域名或 IP： blog.eastmoney.com

随机访问的域名或 IP 实体的恶意值均小于 0.5，且 Web 攻击中的攻击系统和受害系统的恶意值都为 1，符合预期。将攻击路径可视化，恶意值大于 0.5 的恶意实体标为红色，恶意值小于 0.5 的正常实体标为绿色。图 3 所示为场景 1 恶意实体路径可视化，从图中可以看出，攻击实体 112.*.249.56 和受害实体 10.0.0.209 间存在攻击路径，同时受害实体 10.0.0.209 与其他实体也有正常的网络交互。

(2) 场景 2、场景 3

攻击场景 2、场景 3 的分析过程比较相似，因此以场景 2 数据为例进行分析和结果展示。令攻击主机 IP (202.*.1.154) 与获取 bot 文件地址 (202.*.1.63) 为已知初始恶意实体，经过污点传

播模块迭代后，场景 2 的恶意知识图谱子图中所有实体的恶意值结果如表 8 所示。经验证，僵尸网络 bot 主机、传输的恶意文件和受害实体的恶意值均超过阈值。

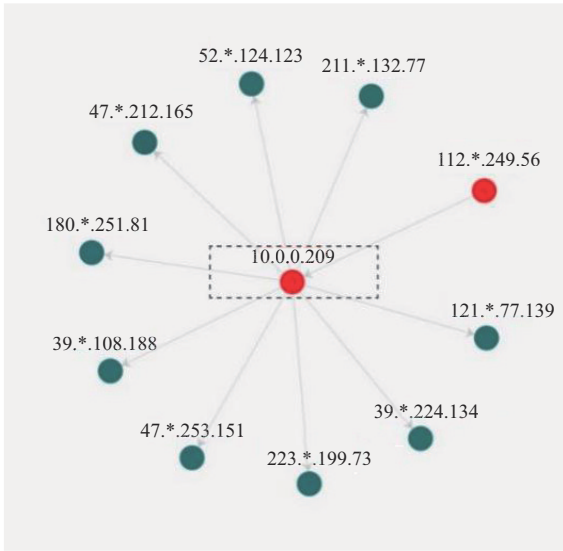


图 3 场景 1 恶意实体路径可视化

Figure 3 Visualization of malicious entity paths in scenario 1

表 8 场景 2 恶意值分布

Table 8 Distribution of malicious values in scenario 2

恶意值区域	实体数量	实体分析
[1]	3	攻击主机 IP：202.*.1.154 获取 bot 文件 IP 地址：202.*.1.63 Botnet Ares 文件：8d49***1695
[0.6, 1)	2	DDoS 攻击受害实体：202.*.7.13 攻击方控制 C&C 服务器：202.*.1.156
[0.5, 0.6)	4	僵尸网络 bot 主机：10.0.0.200~ 10.0.0.203
[0.4, 0.5)	1	僵尸主机访问的实体：202.*.223.182
[0.2, 0.4)	228	日志中正常交互域名或 IP 实体

进一步分析可以发现，各实体恶意值符合预期，如随机访问的域名或 IP 实体的恶意值均小于 0.5，C&C 服务器的恶意值高于僵尸主机，而受 DDoS 攻击的主机恶意值高于僵尸主机。对恶意值大于 0.5 的实体进行可视化展示，如图 4 所示。其中圆形的实体为 IP 地址，方形的实体为域名，菱形的实体为文件，恶意值大于 0.5 的恶意实体标为红色，恶意值小于 0.5 的正常实体标为绿色，恶意值越高对应的红色越深。可视化后

发现, 攻击主机 202.*.1.154 仅与 10.0.0.200 存在信息交互, 即场景中开始的暴力破解 SSH 活动。接着 C&C 服务器 202.*.1.156 和 DDoS 攻击受害实体 202.*.7.13 均与 bot 主机群 10.0.0.200~10.0.0.203 开始信息交互, bot 主机群与 bot 程序分发服务器 202.*.1.63 间也建立了相关关系, 证明了本文知识图谱设计的合理性。

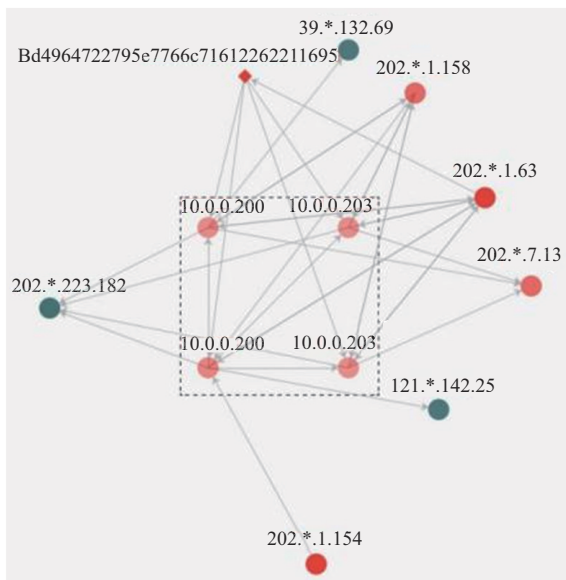


图4 场景2恶意实体路径可视化

Figure 4 Visualization of malicious entity paths in scenario 2

本文还测试了预处理对检测时间的影响, 实验结果如表9所示。Spark 工具的日志并行抽取分析及 Louvain 算法提前划分社区的预处理使污点传播模块的检测时间大大缩短, 进一步证实了本文提出的 CDTP 方法的高效性。

表9 有无预处理检测时间对比

Table 9 Comparison of detection time with and without preprocessing

流量规模/GB	无预处理的检测时间/min	预处理的检测时间/min
9.7	25	3

4.3 权威数据集实验评估

为了进一步证明 CDTP 的可用性, 本文还将 CDTP 方法应用在权威数据集 CIC-IDS2017 和 CSE-CIC-IDS2018 上, 以验证 CDTP 方法能够适应各类攻击流量的检测。

本实验预先标定攻击主机为恶意实体, 为后续

算法提供半监督环境。经过实验, CDTP 方法和信念传播算法在权威数据集上的实验结果如表10所示。

表10 在权威数据集上的实验结果

Table 10 Experimental results on the authoritative dataset

数据集	检测方法种类	准确率	恶意实体召回率	精确率
CIC-IDS2017	CDTP 方法	99.98%	100.00%	85.71%
	信念传播算法	99.55%	83.33%	17.86%
CSE-CIC-IDS2018	CDTP 方法	99.86%	100.00%	91.67%
	信念传播算法	98.64%	72.73%	44.44%

由于 CIC-IDS2017 数据集的规模小于 CSE-CIC-IDS2018, 因此 CDTP 方法和信念传播算法在 CIC-IDS2017 数据集上的表现更出色。另外, 由于两个数据集的恶意实体数量较少, 因此一旦出现误报, 系统的精确率就会迅速下降。尽管如此, CDTP 方法仍然成功检测出了所有的恶意实体。与信念传播算法相比, CDTP 方法在各项检测指标上均展现出显著的性能优势, 表明其设计的优越性。

5 结束语

本文利用 Spark 大数据工具构建知识图谱, 提出了一种名为 CDTP 的基于知识图谱和污点传播的网络攻击检测方法。该方法抽取 IP、域名和文件 3 种实体进行知识图谱构建, 采用 Louvain 社区发现算法在半监督的环境下发现恶意知识图谱子图, 再使用污点传播算法在恶意知识图谱子图上进行网络攻击检测。本文使用自搭建的仿真环境下采集到的数据, 以及权威数据集 CIC-IDS2017 和 CSE-CIC-IDS2018 进行实验, 并将 CDTP 方法与信念传播算法进行对比。结果表明, CDTP 方法在所有实验场景中均能正确检出恶意或受害实体, 并且性能大大优于信念传播算法, 说明本文提出的 CDTP 方法能够从图的角度实现网络攻击检测, 具有较强的实用意义。

尽管本文提出的 CDTP 方法在网络攻击检测中表现出了优越的性能, 但仍有许多领域值得进一步探索和改进。未来的研究可以聚焦于以下几个方面。

1) 随着网络攻击技术的不断演变, 未来的研究需要关注 CDTP 方法如何应对新型或复杂的攻

击模式, 如APT攻击和零日攻击。

2) 研究动态适应机制, 使CDTP方法能够实时更新以应对不断变化的网络攻击策略和技术。

参考文献:

- [1] HNAMTE V, NAJAR A A, NHUNG-NGUYEN H, et al. DDoS attack detection and mitigation using deep neural network in SDN environment[J]. *Computers & Security*, 2024, 138: 103661.
- [2] LEE S Y, WI S, SEO E, et al. ProFiOt: abnormal behavior profiling (ABP) of IoT devices based on a machine learning approach[C]// *Proceedings of the 27th International Telecommunication Networks and Applications Conference (ITNAC)*. Piscataway: IEEE Press, 2017: 1-6.
- [3] 赵文博, 马紫彤, 杨哲. 基于超图神经网络的恶意流量分类模型[J]. *网络与信息安全学报*, 2023, 9(5): 166-177.
- ZHAO W B, MA Z T, YANG Z. Model of the malicious traffic classification based on hypergraph neural network[J]. *Chinese Journal of Network and Information Security*, 2023, 9(5): 166-177.
- [4] CHAU D H P, NACHENBERG C, WILHELM J, et al. Polonium: Tera-scale graph mining and inference for malware detection[C]// *Proceedings of 2011 Society for Industrial and Applied Mathematics International Conference on Data Mining*. 2011: 131-142.
- [5] SUN X, DAI J, LIU P, et al. Using Bayesian networks for probabilistic identification of zero-day attack paths[J]. *IEEE Transactions on Information Forensics and Security*, 2018, 13(10): 2506-2521.
- [6] LYU M, DONG C, CHEN T, et al. A heterogeneous graph learning model for cyber-attack detection[J]. *arXiv preprint arXiv: 2112.08986*, 2021.
- [7] SOSNOWSKI M, SATTler P, ZIRNGIBL J, et al. Propagating threat scores with a TLS ecosystem graph model derived by active measurements[C]// *Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA2024)*. Piscataway: IEEE Press, 2024: 1-11.
- [8] LI A, WANG X, WANG W, et al. A survey of relation extraction of knowledge graphs[C]// *Proceedings of Asia-Pacific Web and Web-Age Information Management Joint International Conference on Web and Big Data*. Berlin: Springer, 2019: 52-66.
- [9] ISMAIL M, ALRABAE S, CHOO K K R, et al. A comprehensive evaluation of machine learning algorithms for Web application attack detection with knowledge graph integration[J]. *Mobile Networks and Applications*, 2024: 1-30.
- [10] MILAJERDI S M, GJOMEMO R, ESHETE B, et al. HOLMES: real-time APT detection through correlation of suspicious information flows[C]// *Proceedings of 2019 IEEE Symposium on Security and Privacy (SP)*. Piscataway: IEEE Press, 2019: 1137-1152.
- [11] NAJAFI P, MÜHLE A, PÜNTER W, et al. Malrank: a measure of maliciousness in SIEM-based knowledge graphs[C]// *Proceedings of the 35th Annual Computer Security Applications Conference*. 2019: 417-429.
- [12] NAJAFI P, PUENTER W, CHENG F, et al. You are your friends: detecting malware via guilt-by-association and exempt-by-reputation[J]. *Computers & Security*, 2024, 136: 103519.
- [13] REN W, SONG X, HONG Y, et al. APT attack detection based on graph convolutional neural networks[J]. *International Journal of Computational Intelligence Systems*, 2023, 16(1): 184.
- [14] QI Y, GU Z, LI A, et al. Cybersecurity knowledge graph enabled attack chain detection for cyber-physical systems[J]. *Computers and Electrical Engineering*, 2023, 108: 108660.
- [15] PAXSON V. Bro: a system for detecting network intruders in real-time [J]. *Computer Networks*, 1999, 31(23-24): 2435-2446.
- [16] PEIMANKAR A, PUTHUSSERYPADY S. DENS-ECG: a deep learning approach for ECG signal delineation[J]. *Expert Systems with Applications*, 2021, 165: 113911.
- [17] DIAKONIKOLAS I, GOEL S, KARMALKAR S, et al. Approximation schemes for ReLU regression[J]. *arXiv preprint arXiv: 2005.12844*, 2020.
- [18] BOURQUIN M, KING A, ROBBINS E. BinSlayer: accurate comparison of binary executables[C]// *Proceedings of the 2nd ACM SIGPLAN Program Protection and Reverse Engineering Workshop*. New York: ACM Press, 2013: 1-10.
- [19] YING R, HE R, CHEN K, et al. Graph convolutional neural networks for web-scale recommender systems[C]// *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York: ACM Press, 2018: 974-983.
- [20] LOW T M, SPAMPINATO D G, MCMILLAN S, et al. Linear algebraic Louvain method in Python[C]// *Proceedings of 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. Piscataway: IEEE Press, 2020: 223-226.
- [21] HE X, LU K, LI G. Taint reverse propagation for analysis of privacy leak[C]// *Proceedings of 2016 IEEE Trustcom/BigDataSE/ICSPA*. Piscataway: IEEE Press, 2016: 1718-1722.
- [22] PEARL J. Probabilistic reasoning in intelligent systems: networks of plausible inference[M]. San Francisco: Morgan Kaufmann, 1988.
- [23] FERRAG M A, MAGLARAS L, MOSCHOYIANNIS S, et al. Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study[J]. *Journal of Information Security and Applications*, 2020, 50: 102419.
- [24] SHARAFALDIN I, LASHKARI A H, GHORBANI A A. Toward generating a new intrusion detection dataset and intrusion traffic characterization[C]// *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*. 2018.

[作者简介]



黄明义 (2000-), 男, 江苏苏州人, 上海交通大学硕士生, 主要研究方向为网络攻防、入侵检测、漏洞挖掘。



周纸墨 (2000-), 男, 黑龙江鹤岗人, 上海交通大学硕士生, 主要研究方向为网络流量检测、入侵检测。



邹福泰 (1973-), 男, 江西吉安人, 上海交通大学高级工程师, 主要研究方向为 DNS 安全、流量安全、网络安全。



张亮 (1974-), 男, 河南安阳人, 国家电网有限公司华东分部正高级工程师, 主要研究方向为电力调度自动化、网络安全。