软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007615] [CSTR: 32375.14.jos.007615]

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

基于流谱的网络流量威胁检测理论及应用*

杨璐铭[1], 王勇军[1], 柳 林[1], 付绍静[1], 赵宝康[1], 苏金树[2]

[1](国防科技大学 计算机学院, 湖南 长沙 410073)

[2](军事科学院, 北京 100091)

通信作者: 王勇军, E-amil: wwyyjj1971@163.com; 柳林, E-amil: liulin16@nudt.edu.cn

摘 要: 随着网络技术的飞速发展, 频频发生的网络攻击事件, 尤其是高级持续性威胁 (advanced persistent threat, APT), 严重影响着国家安全和社会稳定. 在当前网络流量加密、混淆、伪装技术不断发展的背景下, 网络流量的智能化分析可以有效提高网络威胁的检测能力. 但在处理海量的网络流量数据时, 现有诸多方法仍存在分析复杂度高、模型可解释性弱等问题. 流谱以“域变换”作为总体解决思路, 通过为网络流数据构建更加精确、可分离度更高、可观测性更好的描述空间, 实现对网络行为的高效刻画、表征与分析, 从而有效解决上述问题. 类比原子光谱, 提出一套新的流谱方案. 该方案核心思想是通过将网络流映射到一维谱空间实现对网络行为的具象表征, 并以流谱比对方式检测网络流量威胁, 其中良好的流谱分解器设计是关键. 基于半监督自编码器构造流谱分解器并结合重构、分类任务完成训练, 从而使不同网络行为的谱线分布具有良好的可分性. 该方案在 NSL-KDD、UNSW-NB15 和 CIC-DDoS2019 数据集上进行了验证. 实验结果表明, 所提出的流谱方案对网络威胁行为在实现高准确率的识别的同时可以差异化表征对不同网络流量行为, 使得网络行为的可观测性显著提高, 从而增强检测方法的可解释性. 因此, 所提出的流谱方案对网络流威胁行为检测是有效的.

关键词: 流谱; 威胁检测; 网络流; 低维表征; 自编码器

中图法分类号: TP393

中文引用格式: 杨璐铭, 王勇军, 柳林, 付绍静, 赵宝康, 苏金树. 基于流谱的网络流量威胁检测理论及应用. 软件学报. http://www.jos.org.cn/1000-9825/7615.htm

英文引用格式: Yang LM, Wang YJ, Liu L, Fu SJ, Zhao BK, Su JS. Detection Theory and Application Based on Flow Spectrum for Network Traffic Threat. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7615.htm

Detection Theory and Application Based on Flow Spectrum for Network Traffic Threat

YANG Lu-Ming[1], WANG Yong-Jun[1], LIU Lin[1], FU Shao-Jing[1], ZHAO Bao-Kang[1], SU Jin-Shu[2]

[1](College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China)

[2](Academy of Military Science, Beijing 100091, China)

Abstract: With the rapid development of network technology, frequent cyber attacks, especially advanced persistent threats (APTs), seriously affect national security and social stability. Under the continuous evolution of encryption, obfuscation, and camouflage techniques, intelligent analysis of network traffic is considered an effective means to improve threat detection capability. However, when processing massive volumes of network traffic data, existing methods still suffer from high analysis complexity and weak model interpretability. Flow spectrum adopts domain transformation as a unified solution by constructing a more accurate, highly separable, and observable description space for network flow data, thereby enabling efficient characterization, representation and analysis of network behaviors and effectively addressing the above issues. Inspired by the atomic spectrum, this study proposes a novel flow spectrum scheme. The core idea is achieve a concrete representation of network behaviors by mapping network flows into a one-dimensional spectral space,

* 基金项目: 国家自然科学基金 (62072446, 62072431, 62102430, 62372462); 国防科技大学自主科研项目 (ZK22-50)
本文的返修与定稿工作由第一作者在现工作单位军事科学院完成, 并得到现单位有力支持; 本文工作得到了郭世泽院士的方向性指导.
收稿时间: 2025-06-18; 修改时间: 2025-11-20; 采用时间: 2025-12-15; jos 在线出版时间: 2026-04-22

and to detect network traffic threats through flow spectrum comparison, in which the design of an effective flow spectrum decomposer is crucial. In this study, the flow spectrum decomposer is constructed based on a semi-supervised autoencoder and is trained by jointly performing reconstruction and classification tasks, enabling spectral line distributions of different network behaviors to exhibit strong separability. The proposed scheme is validated on the NSL-KDD, UNSW-NB15, and CIC-DDoS2019 datasets. Experimental results show that the proposed scheme achieves high detection accuracy for network threat behaviors while providing differentiated representations for various network traffic behaviors, significantly enhancing network behavior observability and improving the interpretability of threat detection methods. Therefore, the proposed flow spectrum scheme is effective for network traffic threat detection.

Key words: flow spectrum; threat detection; network flow; low-dimensional representation; auto-encoder

当前我国网络规模正在快速发展, 频频发生的网络攻击事件严重影响国家安全和社会稳定. 目前, 网络攻击活动呈现组织化程度高、攻击目标明确、攻击数量增多、隐蔽性增强、攻击效率提高等趋势[1]. 如何从大规模网络流量中及时准确地发现网络威胁行为, 尤其是高级持续性威胁 (advanced persistent threat, APT), 是一项重要的基础性研究. 网络流量是网络行为的主要载体, 具有相同五元组 (源 IP 地址、目的 IP 地址、源端口号、目的端口号、协议类型) 的数据包所构成的网络流是网络分析的基本单元. 针对现实中海量的网络流量数据, 准确、高效、可理解的网络流量分析技术对提高我国网络空间威胁防御能力具有重要意义.

在当前网络流量加密、混淆、伪装技术不断发展的背景下, 以端口过滤和深度包检测 (deep packet inspection, DPI) 为代表的传统基于规则的方法在适用性上已逐渐降低[2,3]. 随着人工智能技术的飞速发展, 智能化的网络流量分析逐渐成为主流[4,5]. 诸多机器学习和深度学习领域的分类及异常检测方法的应用有效提高了网络威胁流量的检测能力[6]. 然而, 当前方法仍存在如下问题.

① 分析过程复杂: 当前网络流量分析仍停留在对高维特征或原始流量的分析阶段, 冗余信息多、分析复杂度高、效率低. 随着流量加密技术的广泛应用, 网络流量的可观测性越来越弱. 这使得网络流量分析工作具有较高门槛, 非专业人员很难直接从复杂的网络流量数据直观理解到网络行为信息.

② “黑箱”问题: 当前的网络威胁检测方法所依赖的机器学习或深度学习模型虽然对一些恶意网络行为可实现较高的检测准确率, 但其不可见且难解释的决策过程提供给网络分析人员和决策者的有价值参考信息有限. 而且现有方法大多只关注分析准确性, 存在“能识别, 难解释”的问题, 其对模型内在决策机理分析的缺失影响了分析模型的可信度.

为了应对国家网络空间大规模数据流观测和不断涌现的网络威胁对抗防御重大需求, 针对传统网络流量威胁监测方法存在的诸多不足, 提出了流谱 (flow spectrum)[7–10]. 流谱借鉴物理学中频谱、光谱方法, 围绕“域变换”作为总体解决思路, 是一种通过对网络流数据进行域变换实现对网络行为的高效刻画和表征的网络流量分析思想. 流谱的核心是构建网络行为的描述空间, 形成可分离、可观测、可解释、可计算的特征表示集合[8,9], 以达到对网络行为刻画更加精确、数据可分离度更高、可表征性更好的效果, 实现对网络流的高效分析[9]. 本文是在该流谱思想下的一次具体技术实现.

本文流谱分析框架是建立在网络空间与原子物理的跨领域类比之上. 在原子物理中, 核外电子能级跃迁这一内部微观机制可以通过原子光谱 (atomic spectrum) 在宏观上形成的稳定、可观测的外部表征, 从而使得通过光谱来反推原子结构成为可能. 而在网络空间中, 网络流由行为意图与通信协议构成的内部机制共同决定, 并可以外化为具有统计规律的流量特征 (即“谱特性”), 从而使通过流谱分析网络流量行为成为可能. 原子光谱与流谱的生成过程示意如图 1 所示. 上述类比的核心思想是将复杂的内部结构信息通过“域变换”转化为易于分析的外部表征信息, 该思想在信号处理、图像识别等诸多领域已被验证有效. 具体在网络流量分析领域, 网络流数据的“谱特性”包括网络行为内在意图的稳定性、行为模式的可分性和特征信息的可压缩性. 通过分析不同网络流行为相似性与大量网络流特征分布数据可以说明网络流数据中存在“谱特性”, 其可被用于网络流的具象表征以实现对网络行为整体特征的高效刻画[7].

面向网络流威胁行为检测任务, 本文提出了一个新的流谱实现方案. 该方案通过将网络流数据从高维空间映射到一维谱空间来表征网络流行为, 从而将复杂的网络流量分析过程简化为流谱分析过程. 相比于传统基于单网

络流的识别检测, 流谱更加关注网络流集合整体的特性, 使得对网络行为的表征具象化. 通过细粒度分析谱线转化过程, 可在网络流的诸多特征中探索对识别检测起主要作用的隐含因素.

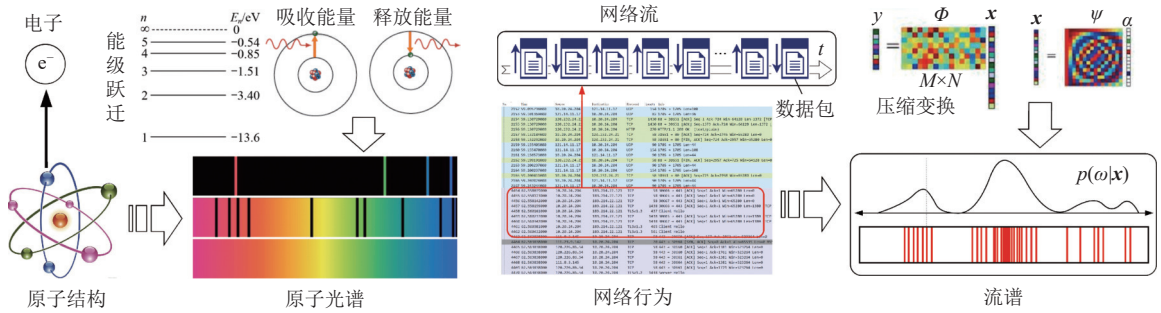


图 1 原子光谱与流谱的生成过程对比

具体而言, 本文的主要工作如下.

① 提出一套新的流谱理论方案, 该方案核心思想是通过将网络流映射到一维谱空间实现对网络行为的具象表征, 并以流谱比对方式实现对网络威胁行为的高效检测.

② 基于半监督自编码器构造流谱的分解器, 并结合数据重构、类型分类、行为对比任务实现了对分解器的训练, 使得不同网络行为的谱线分布具有良好的可分性.

③ 实验验证在 NSL-KDD、UNSW-NB15 和 CIC-DDoS2019 这 3 个网络入侵检测数据集上进行. 实验结果表明, 本文提出的方案不仅可以实现良好的检测准确率, 还可以对不同网络威胁行为实现基于流谱的差异化表征. 实验结果证明了本文提出的流谱方案的有效性.

本文第 2 节主要介绍网络流量威胁检测相关工作. 第 3 节详细阐述一种新的流谱理论的实现方案. 第 4 节具体介绍基于半监督自编码器的流谱分解器的基本设计. 第 5 节介绍了实验设置的基本情况. 在第 6 节中, 通过一系列实验对本文提出流谱方案的有效性进行验证. 第 7 节对本文提出流谱方案的优势和局限性进行讨论. 最后一节对文章内容进行总结.

1 相关工作

目前, 基于机器学习的入侵检测方法已经被广泛研究, 其可以实现比传统的基于规则/签名的入侵检测系统 (network intrusion detection system, NIDS) 更优的检测效果[11]. 从网络流中提取特征并应用机器学习或深度学习模型是目前该领域的主流方法. 当前网络入侵检测的诸多方法主要可分为基于有监督分类的方法与基于异常检测的方法两大类, 本节将从这两个维度梳理研究进展, 并指出现有工作的局限性.

1.1 基于有监督分类的方法

基于有监督学习的分类算法可用于识别恶意流量[5], 现已有诸多相关工作[12−16]. Pajouh 等人[14] 提出了一种基于朴素 Bayes 算法、K 近邻算法和线性判别分析的双层分类模型来提高检测率. Li 等人[15]建立了多类回归模型, 使集成学习适应不同的攻击. Diallo 等人[16]提出一种基于自适应聚类的入侵检测方法, 以解决模型由于对网络流量特征细微变化的敏感性而导致错误分类这一关键挑战. 该方法依赖轻量级神经网络模型学习低维嵌入, 通过自适应聚类和引入多核网络来实现对不同类别网络流量的分离. Barradas 等人[17]提出的 FlowLens 通过数据平面交换机实时线速收集数据包分布特征, 并为每个流计算内存高效的相关特征表示, 从而直接在交换机上实现分类. Zebin 等人[18]基于平衡和堆叠随机森林设计了一个可解释的人工智能模型来准确地检测恶意 DoH (DNS over HTTPS) 流量.

随着深度学习技术不断发展, 诸多基于神经网络的分类方法也被应用于恶意流量分析[19−23]. Wang 等人[19] 在

2017 年首先提出了应用卷积神经网络 (convolutional neural network, CNN) 实现对网络流量的端到端表征学习方案. 其在恶意网络流量分类任务中, 将网络流负载根据字节值转化成了二维图片的格式, 之后使用二维卷积神经网络 (2D-CNN) 实现分类. Sood 等人[20]提出了一种基于深度学习的两阶段降维的 5G 网络边缘入侵检测方法, 其首先使用深度自动编码器来减少特征向量的大小, 并使用 DNN 模型来促进异常检测过程. Min 等人[21]应用词嵌入和文本卷积神经网络 (Text-CNN) 从有效载荷中提取有效信息并进行分类. Corsini 等人[22]提取了表示恶意活动模式的网络流时间序列, 然后对基于 LSTM 的顺序学习模型和传统静态学习模型的有效性进行了比较. Hu 等人[23]提出了一种名为 Graph2vec+RF 的方法, 该方法从网络流的初始几个交互包中构建流图, 并采用图嵌入技术自动化学习流图的向量表示, 之后使用随机森林 (random forest, RF) 对图向量进行分类, 从而实现网络入侵检测.

基于有监督分类的方法虽然可实现高精度, 但它们的分析仅限于已知的标签空间, 无法识别未知类型的恶意流量. 此外, 数据标注也是一个非常昂贵的过程, 需要大量的人力和专家知识的投入[24]. 为缓解网络流量入侵检测领域高质量有标签数据少的挑战, 目前已有工作尝试从元学习 (metalearning) 和半监督学习 (semi-supervised learning) 等角度开展研究.

1.2 基于异常检测的方法

基于异常检测的方法的主要思路是通过学习一类流量的表达表示来训练检测模型, 并通过对模型的适应性来检测未知恶意流量. 文献 [25] 提出的 DiFF-RF 是一种由随机划分二叉树组成的半监督集成学习方法, 专门用于网络入侵检测. Fu 等人[26,27]利用网络流的频域信息作为输入特征, 使用统计聚类算法来检测恶意流量. HyperVision[28]应用无监督图学习方法, 通过分析图的连通性、稀疏性和统计特征来检测加密恶意流量的未知行为模式. Zhang 等人[29]面向 SD-WAN 网络提出了一个名为 OADSD 的异常流量检测框架, 该框架使用分布式动态特征提取从原始流量中提取特征, 并提出利用专家反馈更新参数的按需进化隔离森林使系统具有环境适应能力. Nguyen 等人[30]提出了一种在 Grassmann 流形上结合 PCA 的联邦学习方法, 以学习对物联网设备上良性网络流量的描述.

深度学习的发展使构建更加强大的网络入侵检测模型成为可能. 由于在异常检测方面具有良好的性能, 基于自编码器 (auto-encoder, AE) 的无监督异常检测方法被广泛用于恶意流量检测[31,32]. Zong 等人[33]提出了一种名为 DAGMM 的无监督异常检测方法, 其以端到端的方式联合优化了深度自编码器和高斯混合模型, 设计了独立的评估网络结构来促进混合模型. Mirsky 等人[34]提出了一种名为 Kitsune 的网络入侵检测方法, 其可以在无监督的情况下有效地以在线方式区分正常和异常流量模式. 这是第 1 个在只有良性流量可访问的情况下利用集成自编码器进行入侵检测的工作. 变分自编码器 (variational autoencoder, VAE) 也被应用于未知恶意流量的检测[35–37]. Yang 等人[38]基于条件变分自编码器 (conditional variational autoencoder, CVAE) 和极值理论 (extreme value theory, EVT) 开发了一种智能 NIDS, 能够对已知攻击进行分类并推断未知攻击. 考虑到网络流和自然语言之间数据结构的相似性, 研究人员还将自然语言处理领域的常用方法应用于 NIDS, 如文本卷积神经网络 (Text-CNN) 和门控循环单元 (gate recurrent unit, GRU). Tang 等人[39]提出了一种名为 ZeroWall 的无监督方法, 该方法使用编码器-解码器递归神经网络来训练自翻译机, 以捕获良性请求的语法和语义模式. Kim 等人[40]使用基于双向循环神经网络的 Seq2Seq 序列模型对数据包序列进行建模, 并以自监督训练方式更好的处理类网络入侵检测数据中类不平衡问题, 使得该模型能够在网络攻击的早期阶段及时识别网络攻击行为. Fu 等人[41]基于 U-Net 网络架构来分析数据包长度模式的语义特征, 从而实现对加密隧道下网络攻击流量的检测.

目前, 对比学习、自监督学习和增量学习等新的机器学习范式也开始被用于检测恶意网络流量. Xu 等人[42]提出了一种基于元学习框架的网络入侵检测方法, 并在其中设计了一个名为 FC-Net 的深度神经网络. FC-Net 包括特征提取网络和比较网络两部分. 该方法以判断一对网络流样本是否为同一类型作为基本学习任务, 并通过从一对网络流样本中学习用于分类的特征图并进行比较和判断. Li 等人[43]提出了两种基于 BLS 的算法, 用于对已知网络入侵进行分类, 其中包括增量学习和不包括增量学习. Wang 等人[44]开发了基于对比学习的表示学习方法, 以学习更准确的入侵检测.

总的来说, 网络威胁检测领域大多数工作仍集中在对单网络流的检测准确性上, 而对网络行为的表征与可视

化关注较少, 网络入侵检测的观测性仍然较弱. 提升网络流量分析观测性的关键是将网络流从复杂原始空间转换为低维表征空间 (通常小于 3 维). 然而, 大多数现有工作仍然将网络流表示为高维特征向量, 这不利于网络行为的直观可视化. 因此, 分析者仍然难以直接从检测结果中获取目标主机的网络行为特征.

2 基于流谱的网络流量威胁检测理论

本节主要对基于流谱的网络流量威胁检测理论进行系统的阐述. 首先介绍基于模糊集的流谱构造方案; 之后介绍流谱比对理论并给出了其数学推导过程; 为了提高流谱分析效率, 将在最后介绍该流谱方案的矩阵化过程.

首先对网络流量分析相关概念进行定义. 在本文中, 网络流是由五元组定义的双向数据包集合, 而一个网络行为通常会产生多条网络流. 因此, 对分析网络行为即对其产生的网络流集合进行分析.

在本文流谱分析框架下, 高维特征向量是网络流的底层记录形式并构成分解器 (decomposer) 的输入, 其经分解器实现“域变换”后生成一维谱线以表征网络流; 网络行为的表征结果即为诸多谱线形成的分布 (即流谱). 因此, 通过将网络流数据从高维空间映射到一维谱空间, 可以将复杂的网络流量行为分析简化为流谱分析过程. 与单流检测相比, 流谱分析通过刻画谱线分布特性聚焦网络流集合全局特性, 从而实现对网络流量的“行为级分析”.

对于网络流样本 $\boldsymbol{x}$, 设其随机变量为 X; 而网络流集合为 $\mathcal{X}$ 且其为高维空间 $\mathbb{R}^m$ 的子集. 对于网络行为类别, 设其随机变量为 Y, 样本标签为 y; 而网络流类别空间为集合 $\mathcal{Y}=\{\omega_0,\omega_1,\ldots,\omega_K\}$, 其中包括正常行为 ($\omega_0$) 和 K 种威胁行为 ($\omega_1,\omega_2,\ldots,\omega_K$). 对于生成的谱线, 设其随机变量为 F, 谱线集合 (即流谱) 为 $\mathfrak{F}$; 而谱空间 (谱域) 即为有限实数空间 $\mathbb{F}\subseteq\mathbb{R}$. 谱线是网络流数据到谱空间的映射, 本文定义 $d(\cdot)$ 即为从 m 维空间到一维谱空间的映射 $d\colon \mathbb{R}^m\mapsto\mathbb{F}$. 因此, 对于网络流样本数据 $\boldsymbol{x}$, 本文使用 $d(\boldsymbol{x})$ 来表示谱线数据. 由于高维连续空间具有相同的基数 (空间 $\mathbb{R}^m$ 和 $\mathbb{F}$ 的基数均为 $\aleph_1$), 因此可证明存在从 $\mathbb{R}^m$ 到 $\mathbb{F}$ 的双射.

2.1 基于模糊集的流谱构造方案

在现实中, 针对海量网络流数据来构成有意义的从网络流高维空间 $\mathbb{R}^m$ 到一维谱空间 $\mathbb{F}$ 的双射是困难的, 即不同网络流有可能通过映射 d 生成相同的谱线. 考虑到这种情况, 因此需要使用模糊集来构造流谱.

传统集合具有确定性、互异性、无序性. 其中确定性表示一种元素只能属于一个集合. 而在模糊集理论中, 一种元素可以同时属于多个集合, 其中一个元素属于某集合的程度使用隶属度来表示. 模糊集通常采用 Zedeh 记法, 其中分母是论域中的元素, 分子是该元素对应的隶属度. 因此, 已知网络行为 $\omega\in\mathcal{Y}$ 的已知网络流集合 $\mathcal{X}_\omega$ 可以表示为:

$$\mathcal{X}_\omega=\frac{\mu_\omega(\boldsymbol{x}_1)}{\boldsymbol{x}_1}+\frac{\mu_\omega(\boldsymbol{x}_2)}{\boldsymbol{x}_2}+\ldots+\frac{\mu_\omega(\boldsymbol{x}_n)}{\boldsymbol{x}_n} \tag{1}$$

由于 $d(\cdot)$ 为双射, 其生成已知流谱可以表示为:

$$\mathfrak{F}(\mathcal{X}_\omega)=\frac{\mu_\omega(\boldsymbol{x}_1)}{d(\boldsymbol{x}_1)}+\frac{\mu_\omega(\boldsymbol{x}_2)}{d(\boldsymbol{x}_2)}+\ldots+\frac{\mu_\omega(\boldsymbol{x}_n)}{d(\boldsymbol{x}_n)} \tag{2}$$

其隶属度定义为指定元素 $\boldsymbol{x}$ 条件下其属于此网络行为 ω 的条件概率, 即:

$$\mu_\omega(\boldsymbol{x})=P\{Y=\omega|X=\boldsymbol{x}\} \tag{3}$$

流谱生成过程如后文图 2 所示. 在本方案中使用的是网络流表征向量 $\boldsymbol{x}$ (通常为网络流特征向量) 来表征原始网络流. 对应网络行为 ω 的网络流集合 $\mathcal{X}_\omega$, 每个元素 $\boldsymbol{x}_i$ 对应一个隶属度 $\mu_\omega(\boldsymbol{x}_i)$. 通过映射 $d(\cdot)$ 将 $\boldsymbol{x}_i$ 从高维表征空间映射到一维谱空间, 生成谱线 $d(\boldsymbol{x}_i)$. 因此, 一组特定行为的网络流集合可以转化为一个谱线集合, 并构成一种谱线分布. 这种谱线分布即为流谱, 可以表征特定网络行为特征.

2.2 流谱比对理论

基于流谱实现对网络行为 (尤其是威胁行为) 的检测, 其核心在于通过流谱比对的方式实现对已知网络行为与待检测网络行为的流谱相似度的比较. 本节将详细阐述流谱比对的过程, 并从数学上说明流谱比对结果的实际意义.

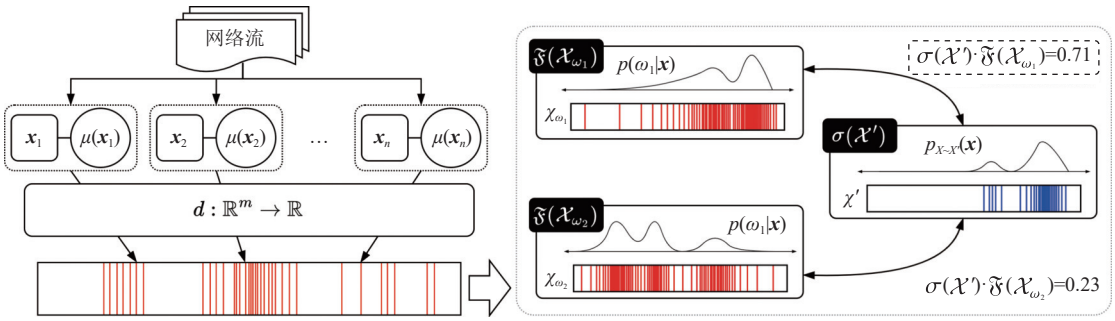


图 2　流谱生成与比对示意图

流谱比对的过程示意如图 2 所示. 图中待检测流谱 $\sigma(\mathcal{X}')$ 分别与已知两个威胁行为谱 $\mathfrak{F}(\mathcal{X}_{\omega_1})$ 与 $\mathfrak{F}(\mathcal{X}_{\omega_2})$ 进行比较, 从而得到检测结果. 从流谱的谱线分布上也可以直观看出, $\mathfrak{F}(\mathcal{X}_{\omega_1})$ 与待检测流谱 $\sigma(\mathcal{X}')$ 更为相似. 经过计算, 两个已知流谱与待检测流谱的比对相似度结果分别为 0.71 和 0.23. 因此, 待检测的网络行为与威胁行为 ω_1 更为相似, 并在该场景下可判定待检测行为是此行为.

谱线隶属度以概率或者概率密度形式进行定义, 是通过谱线出现频次计算得到的. 其谱线隶属度的离散形式为条件概率, 而连续形式则为条件概率密度. 网络流实例 $\boldsymbol{x}$ 的谱线对网络行为 ω 的隶属度 $\mu_\omega(\boldsymbol{x})$ 定义如下:

$$\mu_\omega(\boldsymbol{x}) \overset{\text{离散}}{=} P\{Y=\omega|F=d(\boldsymbol{x})\} \overset{\text{连续}}{=} p(\omega|d(\boldsymbol{x})) \tag{4}$$

而对于待检测网络流集合 $\mathcal{X}'$, 其离散形式的流谱需要计算其谱线分布的概率, 而其连续形式的流谱需要计算其谱线分布的概率密度. 网络流实例 $\boldsymbol{x}$ 的谱线分布概率 $\sigma(\boldsymbol{x})$ 可定义如下:

$$\sigma(\boldsymbol{x}) \overset{\text{离散}}{=} P\{F=d(\boldsymbol{x})|X\in\mathcal{X}'\} \overset{\text{连续}}{=} p_{X\sim\mathcal{X}'}(d(\boldsymbol{x})) \tag{5}$$

对于已知流谱 $\mathfrak{F}(\mathcal{X}_\omega)$ 和待检测流谱 $\sigma(\mathcal{X}')$, 可以通过流谱比对方式实现对未知网络流集合 $\mathcal{X}'$ 行为的判断. 离散形式的流谱比对过程即为两个流谱的乘积, 即:

$$\begin{aligned}\sigma(\mathcal{X}')\cdot\mathfrak{F}(\mathcal{X}_\omega) &= \sum_{\boldsymbol{x}\in\mathcal{X}'}\sigma(\boldsymbol{x})\mu_\omega(\boldsymbol{x}) = \sum_{\boldsymbol{x}\in\mathcal{X}'}P\{F=d(\boldsymbol{x})|X\in\mathcal{X}'\}P\{Y=\omega|F=d(\boldsymbol{x})\}\\ &= \sum_{\boldsymbol{x}\in\mathcal{X}'}P\{X=\boldsymbol{x}|X\in\mathcal{X}'\}P\{Y=\omega|X=\boldsymbol{x}\} = \sum_{\boldsymbol{x}\in\mathcal{X}'}\frac{P\{X=\boldsymbol{x},X\in\mathcal{X}'\}}{P\{X\in\mathcal{X}'\}}\cdot\frac{P\{Y=\omega,X=\boldsymbol{x}\}}{P\{X=\boldsymbol{x}\}}\\ &= \sum_{\boldsymbol{x}\in\mathcal{X}'}\frac{P\{Y=\omega,X=\boldsymbol{x}\}}{P\{X\in\mathcal{X}'\}} = \frac{P\{Y=\omega,X\in\mathcal{X}'\}}{P\{X\in\mathcal{X}'\}} = P\{Y=\omega|X\in\mathcal{X}'\}\end{aligned} \tag{6}$$

离散形式的流谱是最直观朴素的形式, 当已知行为的谱线数据较少且无法形成分布时, 采用离散形式的流谱更为准确和高效. 但是, 离散形式流谱的最大问题是无法处理未知谱线. 具体而言, 由于未知谱线未曾在已知流谱中出现, 故而无法计算其对不同已知网络行为的隶属度.

如果特定行为收集到的网络流足够多, 则其生成谱线可以形成一种分布. 在此情况下, 离散形式流谱比对的计算过程涉及所有已知谱线, 因而计算复杂度较高. 连续形式的流谱不仅可以将比对过程转化成两个分布的相似度比较, 还可以有效处理未知谱线的对于网络行为的隶属度计算问题. 连续形式的流谱比对过程即为两个流谱的卷积, 即:

$$\begin{aligned}\sigma(\mathcal{X}')\cdot\mathfrak{F}(\mathcal{X}_\omega) &= \int_{\mathbb{F}}\sigma(\boldsymbol{x})\mu_\omega(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_{\mathbb{F}}p_{X\sim\mathcal{X}'}(d(\boldsymbol{x}))\,p(\omega|d(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} = \int_{\mathbb{F}}p_{X\sim\mathcal{X}'}(\boldsymbol{x})\,p(\omega|\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \int_{\mathbb{F}}p_{X\sim\mathcal{X}'}(\boldsymbol{x})\,p(\omega|\boldsymbol{x})\frac{p(\omega,\boldsymbol{x})}{p(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} = \int_{\mathbb{F}}p_{X\sim\mathcal{X}'}(\omega,\boldsymbol{x})\mathrm{d}\boldsymbol{x} = P\{Y=\omega|X\in\mathcal{X}'\}\end{aligned} \tag{7}$$

从以上流谱比对过程可以看出, 流谱比对结果是具有实际含义的, 即待检测流谱属于某种网络行为的概率. 在机器学习相关理论中, 该分类结果为分类任务的理论上限, 即 Bayes 最优分类器 (Bayes optimal classifier) 的结果.

对单样本的 Bayes 最优分类器的检测结果为:

$$\hat{y} = \underset{\omega\in\mathcal{Y}}{\operatorname{argmin}} P\{Y=\omega|X=\boldsymbol{x}\} \tag{8}$$

而对于一组网络流 $\mathcal{X}'$ 的检测, 其 Bayes 最优分类器的检测结果就是流谱比对的最大似然估计 (maximum likelihood estimation, MLE) 结果, 即:

$$\hat{y} = \underset{\omega\in\mathcal{Y}}{\operatorname{argmin}} P\{Y=\omega|\boldsymbol{x}\in\mathcal{X}'\} = \underset{\omega\in\mathcal{Y}}{\operatorname{argmax}} \sigma(\mathcal{X}')\cdot\mathfrak{F}(\mathcal{X}_\omega) \tag{9}$$

以上可以说明流谱比对结果具有实际含义, 该方案在理论上具备合理性.

2.3 矩阵化分析

通常情况下, 检测方拥有诸多已知威胁类型及其网络流量数据生成的谱线集合. 在基于流谱的网络威胁检测任务中, 通过生成检测谱矩阵, 可以将流谱比对过程转化为矩阵运算, 从而提高流谱分析的效率.

网络流标签空间 $\mathcal{Y}=\{\omega_0,\omega_1,\ldots,\omega_K\}$ 中包括正常行为 (ω_0) 和 K 种威胁行为. 设谱线统计矩阵为 $\mathbf{N}$, 谱线分布矩阵为 $\mathbf{D}$, 检测矩阵为 $\mathbf{F}$, 待检测流谱的谱向量为 $\boldsymbol{\sigma}$.

对于谱线统计矩阵 $\mathbf{N}=[n_\omega^i]_{\omega\in\mathcal{Y}}$, 其元素 n_ω^i 表示位于 i 且类别为 ω 的谱线数量. 谱线分布矩阵 $\mathbf{D}$ 即是对 $\mathbf{N}$ 的行归一化过程, 其计算过程如下:

$$\mathbf{D}=\left[d_\omega^i\right]_{\omega\in\mathcal{Y}}=\left[\frac{n_\omega^i}{\sum_{j=0}^{K}n_\omega^j}\right]_{\omega\in\mathcal{Y}}^{\mathrm{T}}=\operatorname{diag}^{-1}(\mathbf{N}\mathbf{1})\mathbf{N} \tag{10}$$

其中, $\mathbf{1}$ 表示值全部为 1 的列向量. 谱线分布矩阵 $\mathbf{D}$ 中元素 d_ω^i 表示类别为 ω 谱线中位于 i 的概率. 检测矩阵 $\mathbf{F}$ 是对 $\mathbf{D}$ 的列归一化过程, 计算过程如下:

$$\mathbf{F}=\left[f_\omega^i\right]_{\omega\in\mathcal{Y}}=\left[\frac{d_\omega^i}{\sum_{j=0}^{K}d_\omega^j}\right]_{\omega\in\mathcal{Y}}^{\mathrm{T}}=\mathbf{D}\operatorname{diag}^{-1}\left(\mathbf{1}^{\mathrm{T}}\mathbf{D}\right)\mathbf{D} \tag{11}$$

计算得到的检测矩阵 $\mathbf{D}$ 中元素 f_ω^i 表示位于 i 的谱线属于类别 ω 的条件概率. 因此, 检测矩阵中的元素即为隶属度, 即 $f_\omega^i=\mu_\omega(\boldsymbol{x}_i)$.

不同网络环境下诸多网络行为的分布存在差异, 这种分布差异会影响方法的泛化性能[6]. 为了缓解这种网络行为分布差异造成的影响, 可以根据实际网络环境中不同网络行为的分布情况, 在构造检测矩阵时为网络行为设置权重. 设权重向量 $\boldsymbol{\lambda}=[\lambda_0,\lambda_1,\ldots,\lambda_K]^{\mathrm{T}}$, 则检测矩阵即是对 $\mathbf{D}$ 的列加权归一化过程, 其计算如下:

$$\mathbf{F}=\left[\frac{\lambda_\omega d_\omega^i}{\sum_{j=0}^{K}\lambda_j d_\omega^j}\right]_{\omega\in\mathcal{Y}}^{\mathrm{T}}=\operatorname{diag}(\boldsymbol{\lambda})\mathbf{D}\operatorname{diag}^{-1}\left(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{D}\right)\mathbf{D} \tag{12}$$

对于待检测网络流集合 $\mathcal{X}'$, 其对应待检测谱向量 $\boldsymbol{\sigma}=[p_i]$ 中的元素 p_i 表示 $\mathcal{X}'$ 中所有样本的生成谱线位于 i 的概率. 于是, 可以使用之前生成的检测矩阵来实现识别检测任务. 具体而言, 检测矩阵 $\mathbf{F}$ 与待检测谱向量 $\boldsymbol{\sigma}$ 的乘积即可实现流谱比对过程, 运算结果即为待检测网络流集合属于不同已知网络行为的概率, 最后可以使用最大似然估计方式得到识别结果. 具体计算过程如下:

$$\hat{y}=\underset{\omega\in\mathcal{Y}}{\operatorname{argmin}}\mathbf{F}\boldsymbol{\sigma}=\underset{\omega\in\mathcal{Y}}{\operatorname{argmin}}\operatorname{diag}(\boldsymbol{\lambda})\mathbf{D}\operatorname{diag}^{-1}\left(\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{D}\right)\mathbf{D}\boldsymbol{\sigma} \tag{13}$$

3 基于半监督自编码器的分解器构造

本节主要介绍基于半监督自编码器构造的流谱分解器, 其网络基本结构如图 3 所示. 该分解器基于神经网络结构设计, 并通过多任务训练使其对不同网络行为的生成谱线分布具有良好的可分性. 另外, 本节还对谱域规范化

以及未知谱线处理方法进行介绍.

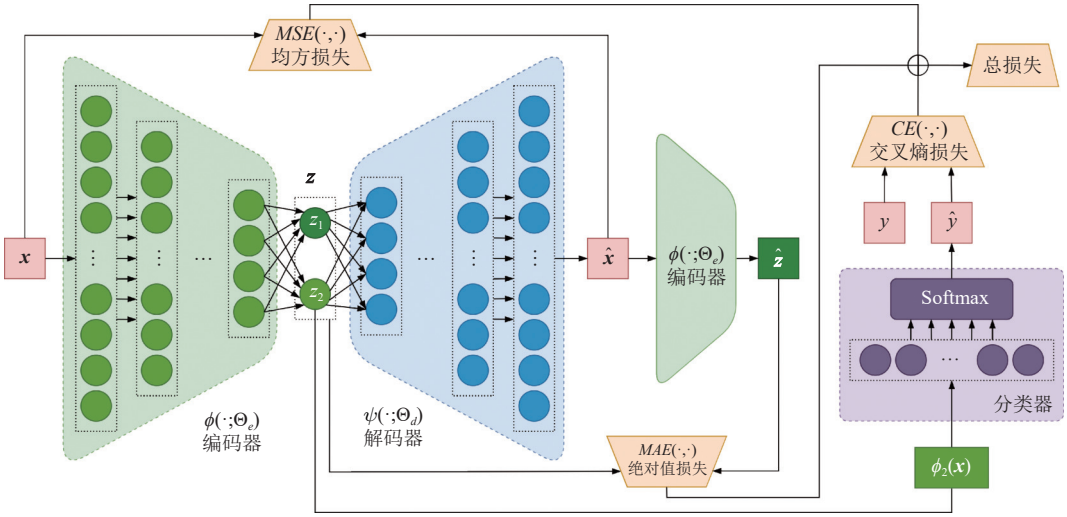


图 3 基于半监督自编码器的分解器网络结构

3.1 分解器网络结构设计

传统机器学习中的降维方法属于通用方法. 而对于流谱来说, 这些传统降维方法可能并非最优选择. 这是因为传统降维方法并非专门为网络流数据而设计的. 因此, 需要寻找尽可能适应网络流数据空间结构的分解器来构造流谱.

自编码器的出现为数据降维重构提供了新的思路. 自编码器 (AE)[45] 是一种人工神经网络结构, 其将输入信息作为学习目标, 从而以无监督的方式对一组数据进行表征学习, 常用于数据降维. 自编码器结构由编码器和解码器两部分构成, 将其分别定义为 $\phi(\cdot;\Theta_e)$ 和 $\psi(\cdot;\Theta_d)$, 其中 Θ_e 和 Θ_d 分别为编码器和解码器的模型参数. 编码器将数据从输入空间映射到编码空间, 而解码器将编码结果再映射回输入空间, 从而实现重构输入数据, 即 $\phi\circ\psi(\boldsymbol{x})\approx\boldsymbol{x}$. 自编码器需要求解映射 ϕ 和 ψ 以使得输入数据重构误差达到最小, 即:

$$\hat{\phi},\hat{\psi}=\underset{\phi,\psi}{\operatorname{argmin}}\|X-\phi\circ\psi(X)\| \tag{14}$$

重输入结构: 传统的自编码器聚焦于对输入数据进行压缩和重构. 然而, 其在编码和解码过程因隐空间缺乏约束而不稳定[46,47]. 为此, 需要引入重输入结构和编码损失以限制二次编码误差, 从而提升隐空间的映射稳定性. 重建样本 $\hat{\boldsymbol{x}}$ 被重新输入编码器 $\phi(\cdot;\Theta_e)$, 得到二次编码结果, 即:

$$\hat{\boldsymbol{z}}=\phi(\hat{\boldsymbol{x}})=\phi\circ\psi\circ\phi(\boldsymbol{x}) \tag{15}$$

其中, 两次编码过程共享相同的编码器参数 Θ_e.

由于自编码器是一种无监督方法, 因而其对网络流数据的重构结果可能无助于网络流量识别分类任务, 甚至可能不利于此. 所以, 在自编码器的求解过程中有必要增加适当的人工知识进行引导. 可以通过多任务学习的方式将分类识别任务引入到自编码器的求解任务中, 这种神经网络结构被称为半监督自编码器 (semi-supervised auto-encoder, semiAE)[48,49]. 多任务学习能够利用任务之间的相似性与差异性, 使得模型可以达到特定的学习效果. 在半监督自编码器中, 分类器是只基于编码器的输出结果而进行的, 但是自编码器和分类器的训练却是同时进行的.

从宏观信息视角看网络流数据, 其中所包含的信息可分为两部分: 可分性信息和不可分性信息. 可分性信息包括但不仅限于具有明显分布区分性的网络流特征, 这些是有助于网络流的分类识别任务的. 不可分性信息为网络流信息中可分性信息之外的部分, 其对网络流数据重构具有重要作用, 但是却无助于网络流的分类识别任务. 考虑到流谱构造的实际需求, 分解器输出结果中可分性信息与不可分性信息的混杂势必会造成生成流谱的区间可分性

降低, 故而需要在分解器构造过程中对网络流的可分性信息与不可分性信息进行分离. 因此, 还需要进一步调整之前设计的自编码器结构. 在本文所提出的方案中, 将自编码器的输出设为两个神经元, 即为 z_1 和 z_2, 其分别代表网络流数据的可分性信息与不可分性信息, 即:

$$z_1, z_2 = \phi(\boldsymbol{x}), \quad \hat{\boldsymbol{x}} = \psi(z_1, z_2) \tag{16}$$

图 3 中的 ϕ_1 和 ϕ_2 分别表示网络结构中编码器部分对输出 z_1 和 z_2 的映射.

本方案使用表示可分性信息的输出 z_2 作为原始谱线值, 其对应映射为 $\phi_2(\cdot)$.

3.2 训练任务与损失函数

分解器的训练过程涉及数据重构和分类任务, 需要设计适当的损失函数. 为控制损失值, 原始数据在输入网络结构之前需要进行预处理, 在本方案中即对数据进行标准化.

数据重构任务是分解器实现将网络流数据从高维空间到谱空间压缩的重要途径, 其主要用于保证流谱构造过程的可逆性. 在本方案中使用均方误差 (mean square error, *MSE*) 来度量分解器的数据重构任务的损失, 其定义如下:

$$\mathcal{L}_{\text{recon}} = \frac{1}{n}\sum_{i=1}^{n} \|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|^2 \tag{17}$$

而对于重输入结构对隐空间的规范化过程, 使用编码损失来衡量原始样本编码结果与重构样本编码结果之间的差异. 具体而言, 该过程使用的损失函数为平均绝对误差 (mean absolute error, *MAE*), 即编码损失为:

$$\mathcal{L}_{\text{enc}} = \frac{1}{n}\sum_{i=1}^{n} |\hat{\boldsymbol{z}}_i - \boldsymbol{z}_i| \tag{18}$$

数据分类任务可以使得分解器将良性行为和攻击行为的网络流数据尽可能映射到谱空间的不同区间, 从而使得不同网络行为的谱线的分布具有显著差异性. 具体而言, 解码器进行数据重构时, 基于的是编码器输出的全部两个神经元 z_1 和 z_2 的输出; 而分类器进行数据分类时, 只是基于表示可分性信息的神经元 z_2 的输出. 通过多任务学习的方式进行多轮次训练, 可以逐步实现网络流数据中可分性信息与不可分性信息的分离, 进而基于表示可分性信息的神经元 z_2 来生成谱线. 在本方案中使用交叉熵 (cross entropy, *CE*) 损失函数来度量分类任务的损失, 其定义如下:

$$\mathcal{L}_{\text{cls}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{\omega\in\mathcal{Y}} y_{i\omega}\log p_{i\omega} \tag{19}$$

其中, $\mathcal{Y}$ 为类别集合, $y_{i\omega}$ 为指示 ω 是否与样本 $\boldsymbol{x}_i$ 所属类别相同的指示变量, $p_{i\omega}$ 为样本 $\boldsymbol{x}_i$ 属于类别 ω 的预测概率.

于是, 最终的损失函数即为自编码器损失和分类器损失的加权和, 其定义如下:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{recon}} + \beta\mathcal{L}_{\text{enc}} + \gamma\mathcal{L}_{\text{cls}} \tag{20}$$

其中, α、β 和 γ 是调节损失权重的参数.

3.3 谱域的规范化

分解器原始输出结果位于实数空间 R 中, 这意味着在整个实数区间都有可能出现谱线. 谱线的分布通常会集中于特定区间, 而在实数区间的±∞两端谱线分布十分稀少. 如果在实数空间 R 构建流谱及检测矩阵时, 需要在无限区间上对谱线分布进行统计分析, 这个过程是复杂且低效的. 因此需要通过寻找适当的一对一映射, 将谱域从无穷区间转化到有限区间中, 从而降低流谱及检测矩阵的复杂度.

双曲正切函数 (tanh) 是从无限区间到有限区间映射的常用方法. 该函数是从 R 到区间 (−1, +1) 的双射, 其函数图像为中心对称的“S”型, 斜率随着自变量绝对值增大而从 1 逐渐趋近于 0. 对位于实数区间的±∞两端的数值, 该函数可将其映射到±1 附近区域. 双曲正切函数这些性质很好地适应谱线分布情况. 对位于±∞两端谱线的稀疏分布, 其可以被双曲正切函数映射到临近±1 区域, 从而避免因无限区间统计困难而造成的谱线丢失; 对于谱线密集分布的中心位置, 双曲正切函数的斜率相对较大, 因而可以基本保留原始谱线分布的特性. 因此, 本方案基于双曲

正切函数实现谱域的规范化.

为了保证谱域规范化前后的谱线分布中心区域的分布性质基本保留, 需要首先对原始谱线值进行标准化处理 (均值为 0, 标准差为 1). 通过统计分析谱线分布, 计算谱线的平均值与标准差, 具体计算方式如下:

$$mean = \frac{1}{N}\sum_{i=1}^{N}\phi_2(\boldsymbol{x}_i)\ std = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\phi_2(\boldsymbol{x}_i)-mean)^2} \tag{21}$$

然后使用双曲正切函数将标准化的谱线值从实数空间映射到有限实数区间. 于是分解器 $d(\cdot)$ 可表示为:

$$d(\boldsymbol{x}) = \tanh\left(\frac{\phi_2(\boldsymbol{x})-mean}{T\cdot std}\right) \tag{22}$$

其中, T 为调节参数, 用于调节双曲正切函数的曲线形状. 因此, 最终得到的谱域 $\mathbb{F}=(-1,1)$.

3.4 对检测中未知谱线的处理

基于流谱的网络行为检测方法实际应用中, 未知流谱中可能会出现在已知流谱中没有出现过的未知谱线. 在离散的谱线分布状态下, 未知谱线由于不存在于已知流谱中, 故而检测矩阵在构造时无法合理计算其对于不同已知网络行为的隶属度并置为 0. 这显然会对基于流谱的检测产生影响. 针对特定行为, 如果收集到的网络流足够多, 根据大数定律, 其生成谱线可以形成一种连续分布. 当待检测流谱中出现未知谱线时, 已知网络行为连续的谱线分布可以根据与其最近邻的已知谱线计算出未知谱线对网络行为的隶属度, 如此可以有效解决对未知谱线的网络行为隶属度计算问题.

核密度估计 (kernel density estimation, KDE) 是常用的未知密度函数估计方法, 属于非参数检验方法. 对于独立同分布的 n 个样本点 $(\boldsymbol{x}_i)$, 其 KDE 拟合的概率密度函数 $\hat{g}(\cdot)$ 如下:

$$\hat{g}(\boldsymbol{x}) = \frac{1}{nh}\sum_{i=1}^{n}\kappa\left(\frac{\boldsymbol{x}}{h},\frac{\boldsymbol{x}_i}{h}\right) \tag{23}$$

其中, $\kappa(\cdot,\cdot)$ 为核函数, h 为平滑参数 (也被称为带宽或窗口). 在 KDE 中最为常用的核函数是 Gauss 核函数, 其又被称为径向基函数 (radial basis function, RBF). 该核函数定义为空间中任意一点到中心点之间 Euclid 距离的单调函数, 其表达式为:

$$\kappa(\boldsymbol{x}_1,\boldsymbol{x}_2) = \exp\left(-\varepsilon^2|\boldsymbol{x}_1-\boldsymbol{x}_2|^2\right) \tag{24}$$

其中, ε 是自由参数. 本文方案中采用基于 Gauss 核函数的 KDE 方法来对谱线分布情况进行拟合.

4 实验设置

本节将主要对实验设置相关情况进行介绍, 包括数据集描述、检测指标、实验环境及参数设置.

4.1 数据集描述

本文在 NSL-KDD、UNSW-NB15 和 CIC-DDoS2019 这 3 个网络入侵检测数据集上对流谱检测效果进行评估, 并基于流谱对不同网络攻击行为进行分析.

NSL-KDD 数据集[50]是 KDD CUP 1999 (KDD99) 数据集的改进版本, 主要用于网络入侵检测研究. 该数据集解决了 KDD99 数据集中的冗余和数据倾斜问题, 优化了训练集与测试集的分布, 使其更适用于机器学习和深度学习方法的评测. 该数据集包含 41 个特征, 涵盖基本特征、内容特征和流量统计特征, 并包含正常流量与 4 大类网络攻击行为 (DoS、R2L、U2R、Probe). 实验按照 2:1 比例划分训练集与测试集.

UNSW-NB15 数据集[51]是由澳大利亚新南威尔士大学 (UNSW) 与澳大利亚国防部联合制作并于 2015 年发布的网络入侵检测数据集, 旨在提供更贴近现实网络环境的流量数据. 该数据集共包含 9 种不同的网络攻击模式, 并且涵盖了来自真实网络环境的流量数据. 其从原始流量中提取了 49 个特征. 实验按照 2:1 比例划分训练集与测试集.

CIC-DDoS2019 数据集[52]是由加拿大网络安全研究院 (CIC) 发布的针对分布式拒绝服务 (DDoS) 攻击检测的

网络流量数据集. 该数据集收集了 2019 年 1 月 22 日和 3 月 11 日两天的网络流量, 正常流量基于 B-Profile 系统生成, 还包括多种类型的 DDoS 攻击行为. 该数据集使用 CICFlowMeter 工具[53,54] 从原始网络流量中提取了 80 多维特征. 实验使用 1 月 22 日的数据进行训练, 并用 3 月 11 日的数据进行测试.

4.2 评价指标

本文实验评价包括两个方面, 包括准确性和可分性两部分. 准确性是对网络流量威胁检测的基础性评估, 实验中将使用准确率、F_1 值、误报率、漏报率、AUC 值对其进行定量评价; 可分性是对基于流谱的网络流量具象表征效果的评估, 实验中将使用轮廓系数和类内类间距离比对其进行定量评价.

网络威胁检测本质上是一个二分类问题. 常见分类准确性指标的相关参数包括: 真正例 (true positive, *TP*)、假正例 (false positive, *FP*)、真负例 (true negative, *TN*)、假负例 (false negative, *FN*). 准确率 (accuracy, *Acc*) 是最常见的评价指标, 表示被正确分类的样本数所占比率, 其定义如下:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{25}$$

精确率 (precision, *P*) 又被称作查准率, 表示识别结果中有多大程度包含在真实数据中. 召回率 (recall, *R*) 又被称作查全率, 反映真实数据有多大程度包含在识别结果中. 精确率和召回率是一对相互矛盾的指标. 为了平衡两者, 常使用其调和平均数 (即 F_1 值) 作为一项综合性指标. 具体而言, 3 个指标的定义如下:

$$P = \frac{TP}{TP+FP},\ R = \frac{TP}{TP+FN},\ F_1 = \frac{2PR}{P+R} \tag{26}$$

误报率 (false alarm rate, *FAR*) 又被称作虚警率、假阳性率, 表示正常样本被错误判定为异常样本的比率. 漏报率 (missing alarm rate, *MAR*) 表示异常样本被错误判定为正常样本的比率. 两个指标定义如下:

$$FAR = \frac{FP}{FP+TN},\ MAR = \frac{FN}{TP+FN} \tag{27}$$

另外, 对于检测任务, 不同阈值设置通常会同时使得诸多指标产生变化, 因此通常绘制 ROC 曲线来评估不同系统的检测性能. ROC 曲线 (receiver operating characteristic curve) 是根据一系列不同的阈值, 以假阳性率 (FPR) 和真阳性率 (TPR) 为横纵坐标绘制的曲线. AUC(area under curve) 值被定义为 ROC 曲线下的面积, 取值范围为 [0, 1]. AUC 值越接近 1, 则检测系统性能越好; AUC 接近 0.5, 检测系统越接近随机.

轮廓系数 (silhouette coefficient, *S*)[55] 通过结合样本的内聚度与分离度来衡量聚类结果中样本归属合理性. 其计算样本 i 的类内样本平均距离 $a(i)$ 与样本到最近其他类的平均距离 $b(i)$, 并通过如下公式计算得到单个样本 i 的轮廓系数值:

$$S(i) = \frac{b(i)-a(i)}{\max\{b(i),a(i)\}}.$$

而轮廓系数即为所有样本该值的平均数. 轮廓系数的值是介于 [−1, 1], 越趋近于 1 代表内聚度和分离度都相对较优. 类内类间距离比 (*D*) 定义为类内平均距离与类间平均距离的比值, 其计算公式如下:

$$D = \frac{\sum\limits_{y_i=y_j}\left|\boldsymbol{x}_i-\boldsymbol{x}_j\right|}{\sum\limits_{y_i\neq y_j}\left|\boldsymbol{x}_i-\boldsymbol{x}_j\right|},$$

该值越小则表明同类样本越聚集且不同类样本越分离. 因而该指标可用于衡量不同网络行为流谱的可分性.

4.3 实验环境及参数设置

实验的运行环境为 Intel(R) Xeon(R) Gold 5222 CPU@3.80 GHz、128 GB RAM 和 NIVIDIA RTX 4090 GPU, 基于 Python 3.9.12 开发实验代码并在 Ubuntu 22.04 系统上进行实验, 深度学习框架为 PyTorch 2.2.1.

在分解器设计中, 本方案使用具有对称结构的多层全连接网络来构造编码器与解码器, 层间激活函数为 tanh 函数. 训练过程中, 本方案设置最终损失函数的权重超参数均为 1, 并使用学习率为 0.001 的 Adam 优化器进行训练. 检测矩阵的构造过程中用于表示不同网络行为分布情况的权重向量被设置为 1.

5 实验分析

基于第 4 节的实验设置, 本节将通过一系列实验对本文提出的流谱方案的有效性进行验证. 本节首先对基于流谱的检测方法的准确性进行测试; 其次对不同类型网络行为的谱线分布情况进行详细的分析; 然后是对分解器处理前后网络流数据的空间可分性的分析; 最后基于流谱对不同网络攻击行为进行谱系相似度分析.

5.1 单流检测准确性分析

本节首先在 3 个网络入侵检测数据集 (NSL-KDD、UNSW-NB15 和 CIC-DDoS2019) 上对流谱分解器的单流检测任务的效果进行了测试, 实验结果如表 1 所示. 在 NSL-KDD 数据集上, 流谱分解器可实现 0.964 0 的准确率和 0.961 8 的 F_1 值; 在 UNSW-NB15 数据集上, 流谱分解器的准确率和 F_1 值均超过 0.98, 其 AUC 值达到了 0.982 3, 其误报率低至 0.000 2; 在 CIC-DDoS2019 数据集上, 流谱分解器不仅在准确率和 F_1 值上高于 0.99, 而且其可以将误报率和漏报率也控制在 0.01 以下. 实验结果表明, 基于半监督自编码器的流谱分解器可以有效进行单流检测任务, 并可以取得良好的检测效果.

表 1　在不同数据集上流谱分解器的检测效果

数据集	准确率 (*Acc*)	F_1 值	误报率 (*FAR*)	漏报率 (*MAR*)	AUC值
NSL-KDD	0.964 0	0.961 8	0.013 8	0.059 8	0.972 6
UNSW-NB15	0.989 1	0.989 1	0.000 2	0.020 5	0.982 3
CIC-DDoS2019	0.994 2	0.995 7	0.004 6	0.006 3	0.998 2

本节之后以 CIC-DDoS2019 数据集为例, 对基于流谱的网络威胁检测的效果分析. 基于流谱的网络行为分析首先需要基于良性网络行为和 DDoS 攻击行为的网络流的谱线分布来构造检测矩阵; 随后对一段时间内的网络行为通过将其谱线分布向量与检测矩阵进行矩阵乘法运算实现检测. 本实验以 10 s 为基本时间间隔, 在特定网段内发起攻击的时间段的每个时间片内产生的网络流集合使用流谱检测矩阵进行检测. 基于本文提出的流谱方法对不同 DDoS 攻击行为的检出率以及检测置信度统计如表 2 所示. 从实验结果中可以看出, 流谱对绝大多数类型的 DDoS 攻击行为可以实现 100% 的检出率. 其中, 对于 LDAP、MSSQL、NetBIOS、SNMP、SSDP、UDP、UDP-lag 和 TFTP 这 8 种 DDoS 攻击行为的流谱检测平均置信度达到了 0.99 以上. 该实验结果表明, 本文所提出的基于流谱的网络威胁检测方法是有效的.

表 2　CIC-DDoS2019 数据集中流谱对不同 DDoS 攻击行为的检测效果

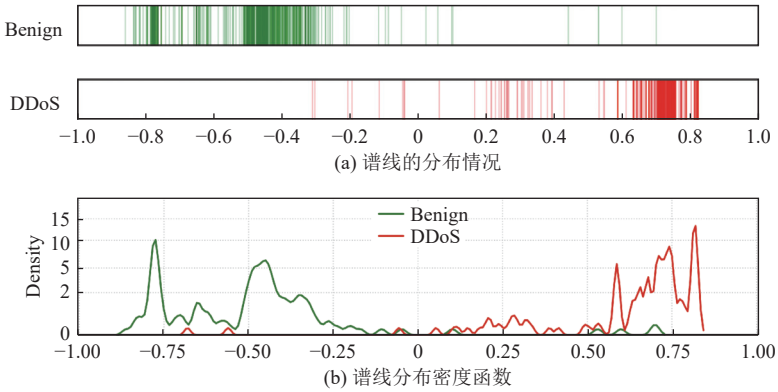
时间段	攻击类型	持续时间 (min)	有效时间片	检出个数	检出率 (%)	置信度
10:35–10:45	NTP	11	52	52	100.00	0.9457±0.0332
10:52–11:16	DNS	25	66	66	100.00	0.9853±0.0275
11:22–11:32	LDAP	11	56	56	100.00	1.000±0.0000
11:36–11:47	MSSQL	12	59	59	100.00	0.9989±0.0001
11:50–12:00	NetBIOS	11	56	56	100.00	0.9971±0.0021
12:12–12:23	SNMP	12	59	59	100.00	0.9992±0.0045
12:27–12:37	SSDP	11	60	60	100.00	0.9989±0.0002
12:45–13:04	UDP	20	55	55	100.00	0.9989±0.0002
13:05–13:15	UDP-lag	11	20	20	100.00	0.9909±0.0395
13:18–13:29	WebDDoS	12	3	2	66.67	0.6669±0.4711
13:30–13:34	SYN	5	28	28	100.00	0.9889±0.0051
13:35–14:05	TFTP	31	131	131	100.00	0.9991±0.0002

5.2 不同网络行为的谱线分布分析

基于流谱, 本节将对不同网络行为流谱中的谱线分布情况进行分析, 并将基于 4 种常用降维算法的分解器作为基线方法与本文方法进行对比分析.

本节首先对 CIC-DDoS2019 数据集中不同网络行为的流谱进行分析. 图 4 展示的是正常网络行为和 DDoS 攻

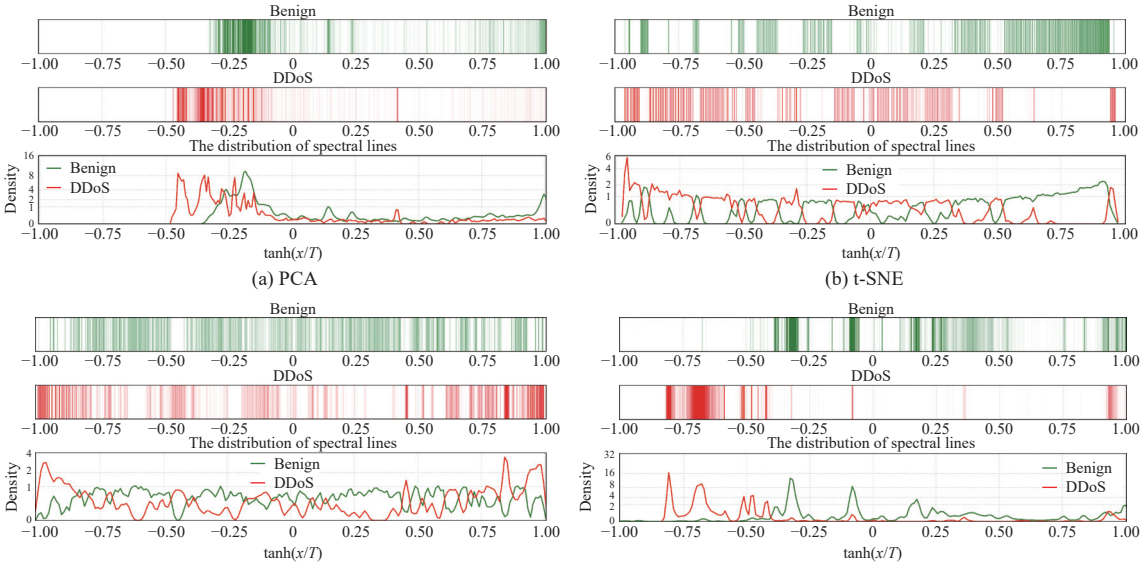
击行为的谱线分布情况. 从图 4 中可以看出, 正常网络行为的网络流所生成的绝大多数谱线均分布在 [−1, 0] 区间内, 而 DDoS 攻击行为的网络流所生成的谱线大多分布在 [0, 1] 区间内, 两种网络行为的谱线分布具有明显的差异. 图 5 展示的是正常网络行为和 DDoS 攻击行为的谱线分布密度函数, 其可以更加清晰的展示谱域中具体位置的谱线分布概率密度. 正常网络行为的谱线主要分布在 [−0.8, −0.2] 区间内, 其与谱线主要分布在谱域正值区域的 DDoS 攻击行为存在着明显差异.



(a) 谱线的分布情况

(b) 谱线分布密度函数

图 4 CIC-DDoS2019 数据集中良性行为和 DDoS 攻击行为的流谱



(a) PCA (b) t-SNE

(c) UMAP (d) 自编码器

图 5 CIC-DDoS2019 数据集上基于现有降维方法的流谱可视化效果

然后, 本节将基于半监督自编码器实现的流谱与使用现有降维方法实现的流谱进行对比. 对于高维向量数据, 常用的降维与可视化方法包括 PCA[56]、t-SNE[57]、UMAP[58]和自编码器[45]. 其中, PCA 是经典的线性降维算法, 而 t-SNE 和 UMAP 属于常用于高维数据可视化的非线性流形学习算法, 而自编码器则是基于神经网络结构的非线性降维方法.

使用上述 4 种降维算法实现的流谱可视化效果如图 5 所示. 流谱的可分性包括良好的类内聚集性和类间分离性. 通过与图 4 的实验结果对比可以看出, 本文基于半监督自编码器设计的分解器可以使得网络流在一维谱空间具有更强的可分性. 具体而言, 对于 t-SNE 和 UMAP 两种流形学习方法所形成的流谱 (图 5(b) 和图 5(c)), 其良性

行为和攻击行为的谱线均分散分布在整个谱空间, 仅在部分谱区间内呈现弱聚集性分布, 其类内聚集性较差. 对于 PCA 算法所形成的流谱 (图 5(a)), 其良性行为和攻击行为的谱线均呈现出了良好的类内聚集性, 但是两者的谱线集中分布区间存在较多重合, 其类间分离性较差. 如图 5(d) 所示, 自编码器模型所形成的谱线通常呈现出多点聚集性分布, 虽然其良性行为和攻击行为的谱线分布区间呈现出较好的类间分离性, 但是相比于基于半监督自编码器的分解器所形成的流谱 (图 4) 类内聚集性仍较弱.

为了对不同网络行为流谱的可分性进行定量分析, 本节将使用轮廓系数和类内类间距离比两个指标来度量该可分性. 在 NSL-KDD、UNSW-NB15 和 CIC-DDoS2019 这 3 个数据集上, 使用不同方法生成流谱的可分性指标结果如表 3 所示. 其中, 加粗数值表示本文方法相比现有方法取得了最优的效果

表 3　CIC-DDoS2019 数据集中谱线分布情况的定量分析

数据集	方法	PCA	t-SNE	UMAP	自编码器	本文方法
NSL-KDD	轮廓系数	0.3057	0.3211	0.2039	0.3235	**0.4423**
	类内类间距离比	0.6610	0.6190	0.7514	0.7198	**0.5445**
UNSW-NB15	轮廓系数	0.1196	0.1423	0.0367	0.2529	**0.6822**
	类内类间距离比	0.8996	0.8342	0.9758	0.7106	**0.3345**
CIC-DDoS2019	轮廓系数	0.0913	0.2248	0.0688	0.3659	**0.6611**
	类内类间距离比	0.8468	0.7141	0.9364	0.5925	**0.3531**

从实验结果可以看出, 使用线性降维 (PCA) 和流形学习 (t-SNE 和 UMAP) 所实现的流谱可分性较弱. 其中, 在 UNSW-NB15 和 CIC-DDoS2019 数据集上, 基于 UMAP 算法所生成流谱的轮廓系数不足 0.1 而类内类间距离比却超过了 0.9, 基于 PCA 算法所生成的流谱的轮廓系数也仅约 0.1. 实验结果说明传统线型降维和流形学习算法对网络流差异化具象表征任务的适应性不足, 从而导致其流谱生成效果较差. 对于现有方法, 在 NSL-KDD 数据集上, t-SNE 算法可以实现最优轮廓系数而自编码器可实现最优的类内类间距离比; 在 UNSW-NB15 和 CIC-DDoS2019 数据集上, 自编码器在两项指标上均为最优. 因此, 相比于其他方法, 基于自编码器的分解器所生成的流谱通常会具有更好的可分性, 这说明神经网络结构能够更好地适应流谱生成任务. 但是, 本文设计的流谱分解器由于引入了半监督机制与重输入机制, 因此生成流谱的可分性显著优于基于自编码器的分解器. 实验结果也说明了本文流谱分解器重半监督自编码器结构设计的合理性.

接下来, 本节对不同类型的威胁行为的谱线分布进行细粒度分析. 图 6 和表 4 分别展示了 CIC-DDoS2019 数据集中不同 DDoS 攻击行为的流谱及其谱线分布的统计信息.

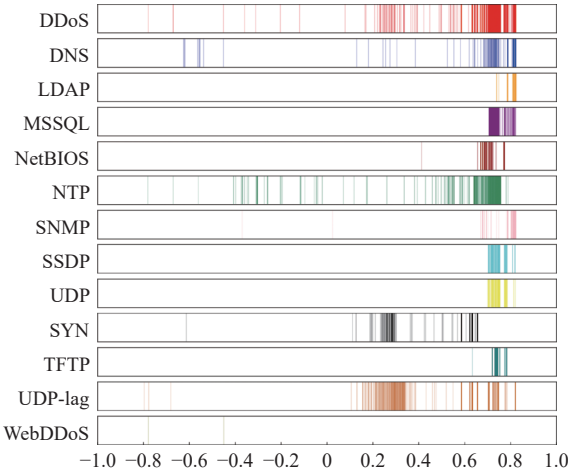


图 6　CIC-DDoS2019 中不同 DDoS 攻击的流谱

表 4 CIC-DDoS2019 中不同 DDoS 攻击的谱线分布

类别	平均值	标准差	Q1	Q2	Q3
正常行为	−0.5396	−0.1786	−0.7659	−0.4826	−0.4404
DNS	0.7904	0.1519	0.8143	0.8144	0.8193
LDAP	0.8148	0.0082	0.8144	0.8144	0.8193
MSSQL	0.7329	0.0409	0.7131	0.7278	0.7408
NetBIOS	0.6887	0.0290	0.6787	0.6787	0.6954
NTP	0.6541	0.2367	0.6531	0.7395	0.7516
SNMP	0.8101	0.0411	0.8142	0.8193	0.8193
SSDP	0.7322	0.0219	0.7070	0.7437	0.7460
UDP	0.7307	0.0208	0.7070	0.7437	0.7460
SYN	0.5787	0.1086	0.5870	0.5870	0.6349
TFTP	0.7309	0.0128	0.7206	0.7323	0.7359
UDP-lag	0.5639	0.1599	0.5870	0.5870	0.6555
WebDDoS	−0.6135	0.1643	−0.6957	−0.6135	−0.5314

注: Q1、Q2、Q3表示谱线集合的四分位数

从图 6 和表 4 中可以看出, 不同网络攻击行为的谱线分布情况之间也具有显著的差异性. 具体而言, MSSQL 攻击行为的谱线集中分布在 [0.7, 0.8] 区间; NetBIOS 攻击行为的谱线集中分布在 [0.6, 0.7] 区间内, 但在 [0.7, 0.8] 中部分位置也有少量集中的谱线分布; NTP 攻击行为的谱线主要分布在 [0.6, 0.8] 区间之内, 但在谱域其他位置也存在少量分散分布; TFTP 攻击行为集中分布在谱线值 0.7323 附近的邻域内. 另外, 从流谱的谱线分布情况可以看出, 一些 DDoS 攻击行为之间存在相似性. 具体而言, DNS、LDAP、SNMP 这 3 种 DDoS 攻击行为均在 [0.81, 0.82] 区间内集中出现了大量谱线; 同样的情况还出现在了 SSDP 和 UDP 两种 DDoS 攻击行为的流谱上, 其均在 [0.7, 0.8] 区间内集中出现了大量谱线; 而 SYN、UDP-lag 两种 DDoS 行为的谱线不仅均在 [0.58, 0.65] 存在谱线的集中分布, 还均在 [0.1, 0.4] 区间内存在谱线的分散式分布. 这些谱线分布的特性与具体 DDoS 攻击的行为特性有关.

另外, 从图 6 中也可以看出, DDoS 攻击行为的谱线分布即为诸多不同具体 DDoS 攻击谱线分布的加和, 这个现象说明本文设计的流谱构造方法满足可加性.

5.3 不同网络威胁行为的谱系分析

对不同类型的网络威胁行为之间的相似性进行分析有助于更好认识网络威胁行为之间的关系. 流谱是一组可以形成稳定分布的谱线所构成的集合, 因此可以通过谱线分布的相似性来定义流谱的相似度. 对于两种网络行为 i 和 j, 其流谱的相似度定义如下:

$$sim(i,j)=1-\frac{1}{2}\int_{\mathbb{F}}\left|g_i(x)-g_j(x)\right|\mathrm{d}x \tag{28}$$

其中, $g_i(\cdot)$ 和 $g_j(\cdot)$ 分别为网络行为 i 和 j 的谱线分布密度函数, $\mathbb{R}$ 为谱域. 由绝对值不等式 $\left|g_i(x)-g_j(x)\right|\geqslant\left|g_i(x)\right|+\left|g_j(x)\right|$ 可知:

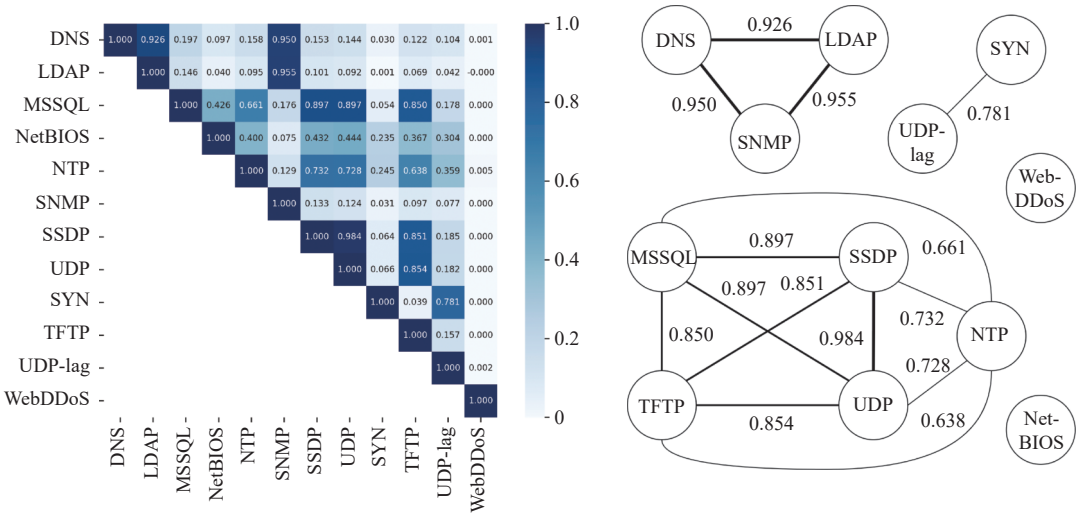
$$sim(i,j)\geqslant 1-\frac{1}{2}\int_{\mathbb{F}}\left(\left|g_i(x)\right|+\left|g_j(x)\right|\right)\mathrm{d}x=1-\frac{1}{2}\left[\int_{\mathbb{F}}g_i(x)\mathrm{d}x+\int_{\mathbb{F}}g_j(x)\mathrm{d}x\mathrm{d}x\right]=0 \tag{29}$$

因此, 相似度函数 $sim(\cdot,\cdot)$ 的值域为 [0, 1].

对于不同网络行为的谱系分析过程中, 首先需使用核密度估计 (KDE) 的方式对不同网络行为的谱线分布进行拟合, 然后计算不同行为两两之间的流谱相似度, 并绘制相似度矩阵以及谱系图.

对于 CIC-DDoS2019 数据集, 图 7 展示了这 12 种网络攻击行为的流谱相似度矩阵 (图 7(a)) 与网络行为谱系图 (图 7(b)). 在绘制该谱系图中, 忽略了流谱相似度低于 0.6 的网络行为相似关系. 从相似度矩阵以及谱系图可以

看出, SSDP 和 UDP 两种 DDoS 攻击行为的流谱相似度最高, 可以达到 0.984; 而 DNS、LDAP 和 SNMP 这 3 种 DDoS 攻击行为之间, MSSQL、SSDP、UDP 和 TFTP 这 4 种 DDoS 攻击行为之间同样具有较高流谱相似度 (两两之间相似度均高于 0.85); UDP-lag 和 SYN 之间的流谱相似度为 0.781, 并显著高于与其他 DDoS 攻击行为的流谱相似度. 以上流谱相似度分析可以说明, 上述这些网络攻击行为在 CICFlowMeter 特征集所能表征的网络行为特性上具有较高的相似性.



(a) 相似度矩阵 (b) 网络行为谱系图

图 7 CIC-DDoS2019 数据集中不同网络威胁行为的流谱相似度分析

5.4 性能分析

本节将对流谱分析性能进行分析. 具体而言, 实验测试了不同输入批次大小下每百万条流的检测时延, 实验结果如表 5 所示.

表 5 流谱分解器每百万条流的时间开销分析 (s)

批次大小 (batch size)	分析时延		差值 (CPU–GPU)
	CPU	GPU	
16	11.8350	17.2782	−5.4432
64	5.9005	6.9628	−1.0623
128	5.2314	5.0201	0.2113
256	4.7270	4.6597	0.0674
512	4.6541	4.4144	0.2398
1024	5.7468	4.5976	1.1492
4096	7.0775	6.0885	0.9890

实验结果表明, 流谱分解器每百万条流的分析时延在使用 CPU 时最低可达 4.6541 s, 而在使用 GPU 时最低可达 4.4144 s. 该检测时延可以应对每秒 10 万条流级别的网络流量实时分析. 从表 5 的实验结果还可以看出, 随着输入批次的增大, 流谱分解器的分析时延呈现先降低再升高的趋势, 且使用 CPU 和 GPU 的分析均在批次大小为 512 左右实现最低的时间开销. 该结果表明, 在实际部署时应当根据硬件性能选择适当的输入批次大小以实现最优的分析性能.

从表 5 中还可看出, 当输入批次较小时, 使用 GPU 的时间开销要大于使用 CPU; 而随着输入批次的增加, 两者差距逐渐缩小并在输入批次大小为 128 时使用 GPU 的时间开销开始优于 CPU 的时间开销. 该结果说明 CPU

适用于处理小批次的流量数据, 而 GPU 对大批次流量数据的处理具有更好的适应性.

6 讨论与局限性分析

本节旨在对本文提出的流谱方案进行深入讨论, 全面审视其理论价值、应用优势与现有不足, 并对未来研究方向进行展望.

当前网络流量威胁分析领域工作大多聚焦于准确率, 对分析过程的可观测性与可解释性等探索有限. 模型提供给分析人员的信息有限, 因而检测结果与对网络行为认知仍存在较大距离. 本文提出的流谱方案通过将网络流数据从高维空间映射到一维谱空间实现了对网络行为的具象表征, 使得分析者无需大量领域知识积累即可通过简单直观的流谱比对实现网络流量行为分析与威胁检测. 基于流谱分析增强了分析过程的可观测性与分析结果的可解释性.

在本文的流谱分析框架下, 分解器的核心功能是“表征”而非“直接检测”, 其输出为表征网络流行为的一维谱线而非分类结果. 基于流谱的检测过程依赖谱线整体分布的比对分析而非模型内部自动化生成的判别阈值. 因此, 训练集不平衡对本方案的影响显著小于传统分类模型. 在网络流量威胁分析领域, 训练集不平衡是常见情况且通常良性样本显著多于恶意样本, 故而流谱分析能够有效适应该场景.

流谱理论为网络流量威胁行为检测提供了新的分析范式, 但本文方案仍然存在一些局限性. 首先, 本方案的流谱分解器输入为高维特征向量, 这导致流谱表征效果依赖于特征工程的有效性. 另外, 本方案虽然相较于分类模型对不平衡训练集不敏感, 但其分解器的训练仍然依赖有标签的流量数据. 未来应当探索适应通用网络环境下基于表征学习的流谱分解器设计, 以进一步提升流谱的分析效率与泛化能力. 未来还应探索流谱在更为复杂的安全场景中的应用, 以持续完善和发展流谱理论.

7 总结与展望

针对现有网络流量行为分析及威胁检测方法存在的分析过程复杂、可解释性弱等问题, 本文提出了一种新的流谱理论方案. 该方案核心思想是通过将网络流映射到一维谱空间实现对网络行为的具象表征并以流谱比对方式实现对网络威胁行为的检测, 其中流谱分解器构造是关键. 本文基于半监督自编码器构造了流谱分解器并结合重构、分类任务完成训练, 从而使得不同网络行为的谱线分布具有良好的可分性. 本文在 NSL-KDD、UNSW-NB15 和 CIC-DDoS2019 这 3 个网络入侵检测数据集上通过一系列实验对该方案的有效性进行验证. 针对网络攻击行为, 该方案可以实现高准确率的识别. 不仅如此, 流谱可以对不同类型的网络攻击行为实现可视化表征, 使得网络行为的可观测性有显著提高, 从而增强检测方法的可解释性.

References

[1] China Cybersecurity Industry Alliance (CCIA). Analysis report on China’s cybersecurity industry (2024). 2024 (in Chinese). https://www.china-cia.org.cn/home/WorkDetail?id=671b5b210200330cd42d1425

[2] Papadogiannaki E, Ioannidis S. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. ACM Computing Surveys (CSUR), 2021, 54(6): 123. [doi: 10.1145/3457904]

[3] Shen M, Ye K, Liu XT, Zhu LH, Kang JW, Yu S, Li Q, Xu K. Machine learning-powered encrypted network traffic analysis: A comprehensive survey. IEEE Communications Surveys & Tutorials, 2023, 25(1): 791–824. [doi: 10.1109/COMST.2022.3208196]

[4] Arp D, Quiring E, Pendlebury F, Warnecke A, Pierazzi F, Wressnegger C, Cavallaro L, Rieck K. Dos and don’ts of machine learning in computer security. In: Proc. of the 31st USENIX Security Symp. (USENIX Security 2022). Boston: USENIX Association, 2022. 3971–3988.

[5] Bertino E, Karim I. AI-powered network security: Approaches and research directions. In: Proc. of the 8th Int’l Conf. on Networking, Systems and Security. Cox’s Bazar: ACM, 2021. 97–105. [doi: 10.1145/3491371.3491384]

[6] Sommer R, Paxson V. Outside the closed world: On using machine learning for network intrusion detection. In: Proc. of the 2010 IEEE Symp. on Security and Privacy. Oakland: IEEE, 2010. 305–316. [doi: 10.1109/SP.2010.25]

[7] Yang LM, Fu SJ, Zhang XY, Guo SZ, Wang YJ, Yang C. FlowSpectrum: A concrete characterization scheme of network traffic behavior

for anomaly detection. World Wide Web, 2022, 25(5): 2139–2161. [doi: 10.1007/s11280-022-01057-8]

[8] Guo SZ, Wang XJ, He MS, Ren CL, Yu SS. Research on intelligent monitoring technology in cyberspace adversarial defense. Information Security and Communications Privacy, 2021(11): 79–94 (in Chinese with English abstract). [doi: 10.3969/j.issn.1009-8054.2021.11.010]

[9] Guo SZ, Lü RJ, He MS, Zhang J, Yu SS. Application of flow spectrum theory in network defense. Journal of Beijing University of Posts and Telecommunications, 2022, 45(3): 19–25 (in Chinese with English abstract). [doi: 10.13190/j.jbupt.2021-271]

[10] Guo SZ, Zhang F, Song ZX, Zhao ZM, Zhao XJ, Wang XJ, Luo XY. Detection of SSL/TLS protocol attacks based on flow spectrum theory. Chinese Journal of Network and Information Security, 2022, 8(1): 30–40 (in Chinese with English abstract). [doi: 10.11959/j.issn.2096-109x.2022004]

[11] Aldweesh A, Derhab A, Emam AZ. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-based Systems, 2020, 189: 105124. [doi: 10.1016/j.knosys.2019.105124]

[12] Dodia P, AlSabah M, Alrawi O, Wang T. Exposing the rat in the tunnel: Using traffic analysis for tor-based malware detection. In: Proc. of the 2022 ACM SIGSAC Conf. on Computer and Communications Security. Los Angeles: ACM, 2022. 875–889. [doi: 10.1145/3548606.3560604]

[13] Yang LM, Fu SJ, Wang YJ, Liang KT, Mo F, Liu B. DEV-ETA: An interpretable detection framework for encrypted malicious traffic. The Computer Journal, 2023, 66(5): 1213–1227. [doi: 10.1093/comjnl/bxac008]

[14] Pajouh HH, Dastghaibyfard G, Hashemi S. Two-tier network anomaly detection model: A machine learning approach. Journal of Intelligent Information Systems, 2017, 48(1): 61–74. [doi: 10.1007/s10844-015-0388-x]

[15] Li XH, Zhu MY, Yang LT, Xu MF, Ma Z, Zhong C, Li H, Xiang Y. Sustainable ensemble learning driving intrusion detection model. IEEE Trans. on Dependable and Secure Computing, 2021, 18(4): 1591–1604. [doi: 10.1109/TDSC.2021.3066202]

[16] Diallo AF, Patras P. Adaptive clustering-based malicious traffic classification at the network edge. In: Proc. of the 2021 IEEE Int'l Conf. on Computer Communications. Vancouver: IEEE, 2021. 1–10. [doi: 10.1109/INFOCOM42981.2021.9488690]

[17] Barradas D, Santos N, Rodrigues L, Ramos FMV, Madeira A. FlowLens: Enabling efficient flow classification for ML-based network security applications. In: Proc. of the 28th Annual Network and Distributed System Security (NDSS) Symp. The Internet Society, 2021.

[18] Zebin T, Rezvy S, Luo Y. An explainable AI-based intrusion detection system for DNS over HTTPS (DoH) attacks. IEEE Trans. on Information Forensics and Security, 2022, 17: 2339–2349. [doi: 10.36227/techrxiv.17696972.v1]

[19] Wang W, Zhu M, Zeng XW, Ye XZ, Sheng YQ. Malware traffic classification using convolutional neural network for representation learning. In: Proc. of the 2017 Int'l Conf. on Information Networking (ICOIN). Da Nang: IEEE, 2017. 712–717. [doi: 10.1109/ICOIN.2017.7899588]

[20] Sood K, Nosouhi MR, Nguyen DDN, Jiang F, Chowdhury M, Doss R. Intrusion detection scheme with dimensionality reduction in next generation networks. IEEE Trans. on Information Forensics and Security, 2023, 18: 965–979. [doi: 10.1109/tifs.2022.3233777]

[21] Min EX, Long J, Liu Q, Cui JJ, Chen W. TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest. Security and Communication Networks, 2018, 2018(1): 4943509. [doi: 10.1155/2018/4943509]

[22] Corsini A, Yang SJ, Apruzzese G. On the evaluation of sequential machine learning for network intrusion detection. In: Proc. of the 16th Int'l Conf. on Availability, Reliability and Security (ARES). Vienna: ACM, 2021. 113. [doi: 10.1145/3465481.3470065]

[23] Hu XY, Gao WJ, Cheng G, Li RD, Zhou YY, Wu H. Toward early and accurate network intrusion detection using graph embedding. IEEE Trans. on Information Forensics and Security, 2023, 18: 5817–5831. [doi: 10.1109/TIFS.2023.3318960]

[24] Li WJ, Meng WZ, Au MH. Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments. Journal of Network and Computer Applications, 2020, 161: 102631. [doi: 10.1016/j.jnca.2020.102631]

[25] Marteau PF. Random partitioning forest for point-wise and collective anomaly detection—Application to network intrusion detection. IEEE Trans. on Information Forensics and Security, 2021, 16: 2157–2172. [doi: 10.1109/TIFS.2021.3050605]

[26] Fu CP, Li Q, Shen M, Xu K. Realtime robust malicious traffic detection via frequency domain analysis. In: Proc. of the 2021 ACM SIGSAC Conf. on Computer and Communications Security. ACM, 2021. 3431–3446. [doi: 10.1145/3460120.3484585]

[27] Fu CP, Li Q, Shen M, Xu K. Frequency domain feature based robust malicious traffic detection. IEEE/ACM Trans. on Networking, 2023, 31(1): 452–467. [doi: 10.1109/TNET.2022.3195871]

[28] Fu CP, Li Q, Xu K. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. In: Proc. of the 30th Annual Network and Distributed System Security Symp. (NDSS). San Diego: The Internet Society, 2023.

[29] Zhang P, He FZ, Zhang H, Hu JK, Huang XH, Wang JL, Yin X, Zhu HH, Li YH. Real-time malicious traffic detection with online isolation forest over SD-WAN. IEEE Trans. on Information Forensics and Security, 2023, 18: 2076–2090. [doi: 10.1109/TIFS.2023.3262121]

[30] Nguyen TA, He JY, Le LT, Bao W, Tran NH. Federated PCA on Grassmann manifold for anomaly detection in IoT networks. In: Proc. of the 2023 IEEE Conf. on Computer Communications (INFOCOM). New York: IEEE, 2023. 1–10. [doi: 10.1109/INFOCOM53939.2023.10229026]

[31] Yousefi-Azar M, Varadharajan V, Hamey L, *et al*. Autoencoder-based feature learning for cyber security applications. In: Proc. of the 2017 Int'l Joint Conf. on Neural Networks (IJCNN). Anchorage: IEEE, 2017. 3854–3861. [doi: 10.1109/IJCNN.2017.7966342]

[32] Yu Y, Long J, Cai ZP. Network intrusion detection through stacking dilated convolutional autoencoders. Security and Communication Networks, 2017, 2017(1): 4184196. [doi: 10.1155/2017/4184196]

[33] Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho DK, Chen HF. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: ICLR, 2018.

[34] Mirsky Y, Doitshman T, Elovici Y, Shabtai A. Kitsune: An ensemble of autoencoders for online network intrusion detection. In: Proc. of the 25th Network and Distributed Systems Security (NDSS) Symp. San Diego: The Internet Society, 2018.

[35] Osada G, Omote K, Nishide T. Network intrusion detection based on semi-supervised variational auto-encoder. In: Proc. of the 22nd European Symp. on Research in Computer Security on Computer Security. Oslo: Springer, 2017. 344–361. [doi: 10.1007/978-3-319-66399-9_19]

[36] Cao VL, Nicolau M, McDermott J. Learning neural representations for network anomaly detection. IEEE Trans. on Cybernetics, 2019, 49(8): 3074–3087. [doi: 10.1109/TCYB.2018.2838668]

[37] Lopez-Martin M, Carro B, Sanchez-Esguevillas A. Variational data generative model for intrusion detection. Knowledge and Information Systems, 2019, 60(1): 569–590. [doi: 10.1007/s10115-018-1306-7]

[38] Yang J, Chen X, Chen SW, Jiang XF, Tan XB. Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection. IEEE Trans. on Information Forensics and Security, 2021, 16: 3538–3553. [doi: 10.1109/TIFS.2021.3083422]

[39] Tang RM, Yang Z, Li ZY, Meng WB, Wang HX, Li Q, Sun YQ, Pei D, Wei T, Xu YF, Liu Y. ZeroWall: Detecting zero-day web attacks through encoder-decoder recurrent neural networks. In: Proc. of the 2020 IEEE Int'l Conf. on Computer Communications. Toronto: IEEE, 2020. 2479–2488. [doi: 10.1109/INFOCOM41043.2020.9155278]

[40] Kim M, Lee D, Lee K, Kim D, Lee S, Kim J. Deep sequence models for packet stream analysis and early decisions. In: Proc. of the 47th IEEE Conf. on Local Computer Networks (LCN). Edmonton: IEEE, 2022. 56–63. [doi: 10.1109/LCN53696.2022.9843272]

[41] Fu CP, Li Q, Shen M, Xu K. Detecting tunneled flooding traffic via deep semantic analysis of packet length patterns. In: Proc. of the 2024 ACMSIGSAC Conf. on Computer and Communications Security (CCS). Salt Lake City: ACM, 2024. 3659–3673. [doi: 10.1145/3658644.3670353]

[42] Xu CY, Shen JZ, Du X. A method of few-shot network intrusion detection based on meta-learning framework. IEEE Trans. on Information Forensics and Security, 2020, 15: 3540–3552. [doi: 10.1109/TIFS.2020.2991876]

[43] Li ZD, Rios ALG, Trajković L. Machine learning for detecting anomalies and intrusions in communication networks. IEEE Journal on Selected Areas in Communications, 2021, 39(7): 2254–2264. [doi: 10.1109/JSAC.2021.3078497]

[44] Wang N, Chen YM, Hu Y, Lou WJ, Hou YT. FeCo: Boosting intrusion detection capability in IoT networks via contrastive learning. In: Proc. of the 2022 Int'l IEEE Conf. on Computer Communications. London: IEEE, 2022. 1409–1418. [doi: 10.1109/INFOCOM48880.2022.9796926]

[45] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504–507. [doi: 10.1126/science.1127647]

[46] Schlegl T, Seeböck P, Waldstein SM, Schmidt-Erfurth U, Langs G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Proc. of the 25th Int'l Conf. on Information Processing in Medical Imaging. Boone: Springer, 2017. 146–157. [doi: 10.1007/978-3-319-59050-9_12]

[47] Akcay S, Atapour-Abarghouei A, Breckon TP. GANomaly: Semi-supervised anomaly detection via adversarial training. In: Proc. of the 14th Asian Conf. on Computer Vision (ACCV). Perth: Springer, 2018. 622–637. [doi: 10.1007/978-3-030-20893-6_39]

[48] Wu HY, Yang HM, Li XM, Ren HJ. Semi-supervised autoencoder: A joint approach of representation and classification. In: Proc. of the 2015 Int'l Conf. on Computational Intelligence and Communication Networks (CICN). Jabalpur: IEEE, 2015. 1424–1430. [doi: 10.1109/CICN.2015.275]

[49] Zhuang FZ, Luo D, Jin X, Xiong H, Luo P, He Q. Representation learning via semi-supervised autoencoder for multi-task learning. In: Proc. of the 2015 IEEE Int'l Conf. on Data Mining. Atlantic City: IEEE, 2015. 1141–1146. [doi: 10.1109/ICDM.2015.22]

[50] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In: Proc. of the 2009 IEEE Symp. on Computational Intelligence for Security and Defense Applications (CISDA). Ottawa: IEEE, 2009. 1–6. [doi: 10.1109/CISDA.2009.

5356528]

[51] Moustafa N, Slay J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proc. of the 2015 Military Communications and Information Systems Conf. (MilCIS). Canberra: IEEE, 2015. 1–6. [doi: 10.1109/MilCIS.2015.7348942]

[52] Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: Proc. of the 2019 Int'l Carnahan Conf. on Security Technology (ICCST). Chennai: IEEE, 2019. 1–8. [doi: 10.1109/CCST.2019.8888419]

[53] Draper-Gil G, Lashkari AH, Mamun MSI, Ghorbani AA. Characterization of encrypted and VPN traffic using time-related features. In: Proc. of the 2nd Int'l Conf. on Information Systems Security and Privacy (ICISSP). Rome: SciTePress, 2016. 407–414.

[54] Lashkari AH, Gil GD, Mamun MSI, Ghorbani AA. Characterization of tor traffic using time based features. In: Proc. of the 3rd Int'l Conf. on Information Systems Security and Privacy. Porto: SciTePress, 2017. 253–262.

[55] Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 1987, 20: 53–65. [doi: 10.1016/0377-0427(87)90125-7]

[56] Hotelling H. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 1933, 24(7): 498–520. [doi: 10.1037/h0070888]

[57] van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9(86): 2579–2605.

[58] McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform manifold approximation and projection. The Journal of Open Source Software, 2018, 3(29): 861. [doi: 10.21105/joss.00861]

附中文参考文献

[1] 中国网络安全产业联盟. 中国网络安全产业分析报告 (2024 年). 2024. https://www.china-cia.org.cn/home/WorkDetail?id=671b5b210200330cd42d1425

[8] 郭世泽, 王小娟, 何明枢, 任传伦, 俞赛赛. 网络空间对抗防御中的智能监测技术研究. 信息安全与通信保密, 2021(11): 79–94. [doi: 10.3969/j.issn.1009-8054.2021.11.010]

[9] 郭世泽, 吕仁健, 何明枢, 张杰, 俞赛赛. 流谱理论及其在网络防御中的应用. 北京邮电大学学报, 2022, 45(3): 19–25. [doi: 10.13190/j.jbupt.2021-271]

[10] 郭世泽, 张帆, 宋卓学, 赵子鸣, 赵新杰, 王小娟, 罗向阳. 基于流谱理论的 SSL/TLS 协议攻击检测方法. 网络与信息安全学报, 2022, 8(1): 30–40. [doi: 10.11959/j.issn.2096-109x.2022004]

作者简介

杨璐铭, 博士, 助理研究员, CCF 专业会员, 主要研究领域为加密流量分析, 网络威胁检测, 人工智能, 机器学习.

王勇军, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为网络威胁行为分析与检测, 软件安全分析.

柳林, 博士, 副教授, CCF 专业会员, 主要研究领域为密码学, 网络安全.

付绍静, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为密码学, 网络空间安全.

赵宝康, 博士, 副教授, CCF 杰出会员, 主要研究领域为计算机网络, 网络安全, 人工智能.

苏金树, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为网络体系结构, 网络空间安全, 智能网络.