软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007682] [CSTR: 32375.14.jos.007682]

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

基于 MCP 智能体的通用威胁情报知识图谱构建方法*

沙乐天, 薛 磊, 陈 霄, 李德强, 肖 甫

(南京邮电大学 计算机学院, 江苏 南京 210023)

通信作者: 薛磊, E-mail: 1223045536@njupt.edu.cn

摘 要: 随着网络攻击的复杂性不断增加, 威胁情报的收集与整合面临着分散化问题. 在此背景下, 企业开始构建私有化的威胁情报知识图谱系统. 然而传统构建方法效率低下, 因实体关系抽取不准确而难以有效发挥作用. 为解决以上问题, 提出一种基于 MCP (model context protocol) 智能体的通用威胁情报知识图谱构建框架 OPFA (one prompt for all). 通过动态语义主体定位, 微调大语言模型的提示词模板, 从而驱动智能体对威胁实体关系进行抽取. 威胁实体关系将作为知识图谱的节点与边被自动化创建, 通过实体属性值匹配以关联威胁情报, 形成完整的知识图谱. 接着, 系统会针对关键威胁实体 (如漏洞、恶意样本) 拉取 MCP 资源从而完成订制化的知识图谱扩充. 实验表明, 智能体不仅能够提升威胁实体和实体关系抽取的精确率 (分别为 97.22% 和 97.83%)、召回率 (分别为 90.91% 和 95.52%) 与 $F1$ 值 (分别为 94.44% 和 96.66%), 还能有效提高知识图谱的构建效率.

关键词: MCP; 智能体; 大语言模型; 威胁情报; 知识图谱; 实体识别; 实体关系

中图法分类号: TP309

中文引用格式: 沙乐天, 薛磊, 陈霄, 李德强, 肖甫. 基于MCP智能体的通用威胁情报知识图谱构建方法. 软件学报. http://www.jos.org.cn/1000-9825/7682.htm

英文引用格式: Sha LT, Xue L, Chen X, Li DQ, Xiao F. Construction Method for Universal Threat Intelligence Knowledge Graph Based on MCP Agent. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7682.htm

Construction Method for Universal Threat Intelligence Knowledge Graph Based on MCP Agent

SHA Le-Tian, XUE Lei, CHEN Xiao, LI De-Qiang, XIAO Fu

(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: With the increasing complexity of cyber attacks, the collection and integration of threat intelligence face challenges of fragmentation. In this context, enterprises have begun building private threat intelligence knowledge graph systems. However, traditional construction methods suffer from inefficiency and limited effectiveness due to inaccurate extraction of entities and their relationships. To address these issues, this study proposes OPFA: a general threat intelligence knowledge graph construction framework based on model context protocol (MCP) agents. Through dynamic semantic subject positioning, large language model prompt templates are fine-tuned to drive agents to extract threat entities and their relationships. These threat entities and relationships are automatically created as nodes and edges in the knowledge graph, and threat intelligence is then linked through entity attribute value matching to form a complete knowledge graph. The system subsequently retrieves MCP resources for critical threat entities (such as vulnerabilities and malicious samples) to achieve customized knowledge graph expansion. Experimental results demonstrate that the agents not only improve *Precision* (97.22% and 97.83%, respectively), *Recall* (90.91% and 95.52%, respectively), and $F1$-score (94.44% and 96.66%, respectively) in threat entity and entity relationship extraction, but also effectively enhance the construction efficiency of the knowledge graph.

Key words: model context protocol (MCP); agent; large language model (LLM); threat intelligence; knowledge graph; entity recognition; entity relation

---

* 基金项目: 国家自然科学基金 (62302236); 江苏省前沿技术研发计划 (BF2024071); 江苏省科学技术厅重点研发计划 (BE2022081)
收稿时间: 2025-04-21; 修改时间: 2025-09-28; 采用时间: 2026-04-07; jos 在线出版时间: 2026-07-15

网络空间威胁情报 (cyber threat intelligence, CTI) 是指通过系统化收集、分析和处理网络攻击相关数据, 形成的可指导安全防御的知识体系. 其目标是帮助组织提前识别风险、理解攻击者行为并制定精准防护策略[1,2]. 企业网络安全防护人员通常会依赖在线 CTI 平台 (如微步、安天等) 获取威胁情报信息. 然而这种单点式的收集方式难以有效提升企业对于网络攻击威胁的整体感知能力. 在此背景下, 知识图谱逐渐成为网络安全防护领域的研究热点. 它通过结构化关联分析和动态推理能力有效解决传统威胁情报孤立、碎片化的问题, 为企业提供更全面、更系统的安全防护支持[3]. 然而, 知识图谱在实际应用中仍面临诸多挑战. 当前知识图谱的构建过程往往需要整合多源数据渠道, 但不同数据来源的异构性使得建模框架的兼容性不足, 导致数据整合难度较大. 此外, 知识图谱的质量评估与标准化程度参差不齐, 进一步加剧了跨企业威胁情报共享的互操作性障碍, 形成分散化的数据孤岛. 该问题不仅限制了威胁情报的共享效率, 也阻碍了网络安全防护能力的整体提升. 因此, 业界迫切需要一种通用且高效的自动化 CTI 知识图谱构建方法, 以解决现有技术瓶颈.

实体和实体关系的抽取是知识图谱构建的关键步骤, 传统方法[4–7]主要面临以下问题: 对于实体和实体关系预定义规则以及人工标注的强依赖性导致其抽取结果虽然在特定语义环境下具有一定的准确性, 但泛化能力较弱. 难以适应复杂多变的 CTI 语义. 目前, 已有研究[8–9]利用基于大语言模型 (large language model, LLM) 的方法从 CTI 报告中抽取实体和实体关系以手动构建知识图谱, 但这些方法本质上依赖于提示词工程. 大语言模型本身主要用于单会话推理任务, 且在静态提示词驱动下的推理和输出模式通常是固定的, 无法实现多个会话交互下的动态上下文管理. 因此, 当面临预定义提示词之外的任务场景时, 大语言模型往往难以准确推理. 而在 CTI 知识图谱构建任务中, 常会遇到新兴威胁下动态变化的语义环境, 这对大语言模型的上下文适应性提出了更高的要求. 模型上下文协议 (model context protocol, MCP)[10–11]作为一种 LLM 协同工作的智能体交互框架, 为上述问题提供了新的解决方案. MCP 通过标准化上下文描述语言和任务分解机制, 实现了 LLM 在开放环境下的可控推理与精准知识调用[12], 这使其具有优秀的上下文工程适配性. 借助 MCP 完成单提示词全流程 CTI 采集与知识图谱构建, 是本文的研究重点. 具体而言, 大语言模型原生的语义感知能力确保了威胁实体与实体关系的基本抽取质量, 这将在交互提示词的驱动下自动完成, 无需预定义规则和数据标签作为约束. 在此基础上, 基于智能体协同框架的模型上下文管理和调整能力所构建的实时消歧与本体对齐通道[13], 能够将抽取结果在短时间内完成语义验证, 并同步映射至知识图谱库, 形成从抽取到验证再到映射的闭环流程. 这一机制不仅实现了威胁情报从非结构化数据到图谱节点的实时同步转化, 还展示了单一自然语言指令驱动 CTI 知识图谱构建流程的典型范式, 为大语言模型在网络安全防护领域的应用提供了全新的技术思路.

为了实现自然语言驱动下低门槛交互式的构建模式, 本文提出一种基于 MCP 智能体的通用威胁情报知识图谱构建框架: OPFA (one prompt for all), 即单提示词-全流程. 基于大语言模型的语义感知和分析能力, 将用户的采集与构建需求转化为 MCP 工具的结构化输入参数, 并通过这些参数调用 MCP 工具执行对应的操作, 最终返回结果. 具体而言, 我们通过引入动态语义主体定位技术, 以协助模型在处理长文本信息时保持专注. 模型识别主体后生成对应的威胁情报实体定义, 通过实体交叉关联生成其实体关系定义, 有效弥补了预定义或无定义情况下模型在实体抽取任务中所表现出的不稳定性. 除此之外, 该工作进一步探索了提示词对大语言模型任务驱动的影响, 通过微调 MCP 提示词模板, 优化了其实体识别与实体关系抽取的准确性, 并有效耦合了工具调用和模型交互过程中产生的参差. 结合这两项技术, OPFA 实现了对智能体交互上下文的动态定义与调整, 从而更好地理解并抽取未知或新兴的威胁实体及关系. 总体而言, 本工作降低了知识图谱构建的技术门槛, 实现了从需求感知到任务交付的全流程自动化, 使非专业人员能够通过自然语言指令触发复杂知识图谱的敏捷构建, 将传统数周的构建周期压缩至小时级, 显著提升了威胁情报知识图谱的构建效率.

1 相关工作

早期, 研究者多聚焦于将知识图谱应用于医疗[14]、电子商务[15]等领域, 而近年来, 知识图谱在网络安全领域的应用价值日益凸显, 逐渐成为研究热点. 研究人员利用知识图谱技术完成了威胁检测、攻击溯源、安全态势感

知等网络安全相关任务[16–19]. 目前, 威胁情报知识图谱的构建方法主要分为基于规则、基于深度学习和基于大语言模型这 3 类, 而这些方法的差异性主要体现在实体识别和实体关系抽取的过程中.

基于规则的方法主要通过预定义的模式匹配来提取威胁情报中的实体和关系[20]. 史慧洋等人[4]提出了一种威胁情报知识图谱框架, 通过情报搜集、信息抽取、本体构建和知识推理等步骤构建知识图谱, 并采用正则匹配机制对输出结果进行限制, 以高效识别和抽取文本信息中的失陷指标. Piplai 等人[5]利用相同方法从恶意软件报告中构建网络安全知识图谱, 有效地构建了一个融合安全知识实体的检索系统. 虽然上述方法在特定场景下具有高准确性和可解释性, 但其局限性也十分显著, 主要体现在其繁琐的规则构建过程以及较差的覆盖与泛化能力.

基于深度学习的方法利用神经网络 (如 LSTM、BERT 等) 自动学习文本特征, 进行实体识别和关系抽取. Mouiche 等人[6]提出了一种基于深度学习的威胁情报知识图谱构建框架, 该框架利用针对网络安全领域优化的 SecureBERT 模型, 并结合基于注意力机制的 BiLSTM, 来捕捉安全文本的上下文和细微差别, 从而减少错误传播并提高提取精度. Priya 等人[7]提出了一种混合深度学习模型, 通过特征选择分析和攻击阶段相关性检测高级持续性威胁, 并在实际场景中验证了其有效性. Böge 等人[21]提出一种混合深度学习架构, 结合 Transformer 和卷积神经网络的优势, 并通过标准化命令语言转换器提高自然语言处理的效率和准确性. Sarhan 等人[22]提出 Open-CyKG 框架, 采用基于注意力的神经式开放信息模型构建知识图谱, 可以有效地规范提取到的结构化数据. 黄智勇等人[23]基于 BiLSTM-CRF 模型引入外部网络安全词典来加强网络安全文本的特征, 并结合多头注意力机制提取多层特征, 最终在网络安全数据集上取得了更优异的结果, 然而该方法在跨语言安全数据提取中的表现不理想.

当下, 大语言模型在从非结构化文本中提取结构化信息方面也展现出了巨大的潜力[24]. Li 等人[25]尝试使用 NLP 模型分析 CTI 报告中的威胁特征, 并在人工标注的数据集中实现了较高的攻击特征识别准确率. Ferrag 等人[26]提出了一种基于 BERT 的轻量级模型, 利用大语言模型的语义处理能力, 结合隐私保护技术, 实现了高效的威胁情报提取和分析. Hu 等人[8]则利用 GPT 的少样本学习能力, 通过设计合适的提示词 (prompt) 进行数据标注和增强, 从而减少人工标注的工作量, 并使用 LoRA 技术对 Llama2-7B 模型进行指令微调, 实现威胁情报文本的主题分类、实体和关系抽取以及 TTP 分类, 并最终构建知识图谱. Liu 等人[9]使用 ChatGPT 自动提取攻击相关的实体及其关系, 验证了其方法在低资源场景下的可行性和适用性. 赖清楠等人[27]通过指令提示和上下文学习, 自动从网络威胁情报中生成网络安全知识图谱和攻击知识图谱, 并提供可操作的防护建议, 延伸了其在网络防御系统中的应用[28].

综上所述, 当前相关研究工作的重点是利用基于深度学习和大语言模型的方法, 在实体识别与实体关系抽取方面, 挖掘威胁情报中的深层次语义信息, 并构建知识图谱. 其核心难点在于实体与实体关系的定义及模型权重参数难以在新型威胁情报中保持迁移性. 因此本工作主要围绕如何使用 MCP 架构串联大语言模型完成实时性对齐工作而展开. 本文提出利用动态主体定位与提示词微调技术来解决迁移性问题, 并在公开威胁情报平台采集的数据集中验证了该方法的有效性.

2 MCP 总体架构

MCP 的出现统一了大语言模型和外部数据源工具之间的通信协议. 其总体架构如图 1 所示. 核心组件包括 MCP 主机、MCP 客户端以及 MCP 服务器. 各组件相互配合, 确保了 MCP 在运行以及管理过程中的资源调用、模型交互以及通信安全[29,30].

(1) MCP 主机作为核心组件, 扮演着至关重要的角色. 它不仅是用户与 MCP 系统交互的入口, 负责接收用户的输入并将其传递给内部的 MCP 客户端, 还承担着管理多个 MCP 客户端实例的任务, 每个实例与特定的 MCP 服务器建立一对一的连接. MCP 主机在协调 AI 或大语言模型的集成和采样过程中发挥着关键作用, 确保模型能够高效地与外部数据源或工具进行交互. 此外, 它还负责上下文聚合, 整合来自多个数据源的信息, 以支持复杂任务的处理. 在与外部数据源交互时, MCP 主机维护着清晰的安全边界, 保障数据访问的安全性和隐私性. 最终, MCP 主机通过与 MCP 服务器的协作, 执行用户请求的任务, 并将结果返回给用户, 从而完成整个交互流程.

(2) MCP 客户端充当大语言模型和 MCP 服务端之间的桥梁. 其首先从服务端获取可用的工具列表, 接着将用户的查询连同工具描述通过方法调用一起发送给大语言模型. 大语言模型决定是否需要使用工具以及使用哪些工

具. 如果确定使用工具, 服务端将通过客户端执行相应的工具调用. 最终大语言模型整合工具调用返回的结果并且生成自然语言响应给用户.

(3) MCP 服务端主要提供 3 种类型的功能. 资源 (resource) 类似文件数据, 可被客户端读取; 工具 (tool) 是可被大语言模型调用的方法; 提示 (prompt) 一般为预定义的提示词模板, 当大语言模型无法继续任务或者通过上下文感知到特定语境时, 会调用该模板从而驱动模型按照提示词继续执行. 这些功能赋予了大语言模型丰富的上下文信息和操作能力, 使其能够自动化地执行整个任务.

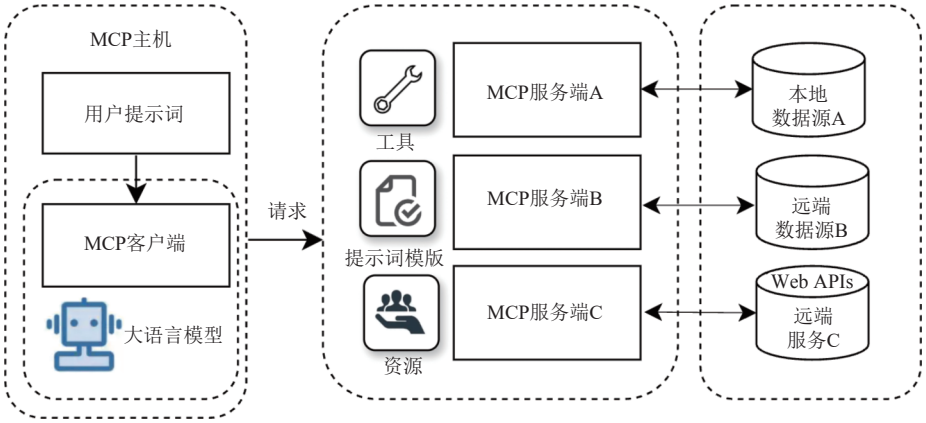


图 1 MCP 框架图

3 OPFA 框架

图 2 展示了 OPFA 框架的整体设计, 该框架符合 MCP 架构标准. 用户在主机侧输入提示词表示自己的需求. 搭载了大语言模型的 MCP 客户端收到用户需求并根据语义推测需要调用的 MCP 工具以及提示词模板. 这里, 模型选择使用 CTI 采集工具去采集规定日期内的情报信息. 接着客户端将工具返回结果存储到 MCP 资源库中. 服务端会动态地从资源库中抽取 CTI 作为提示词模板生成的学习语料, 这个过程涉及使用动态语义主体定位技术去微调由泛化实体定义的提示词. 客户端模型会自动调用该模板进行实体和实体关系的抽取工作. 最终知识图谱构建工具将为每一个实体和实体关系创建对应的点与边.

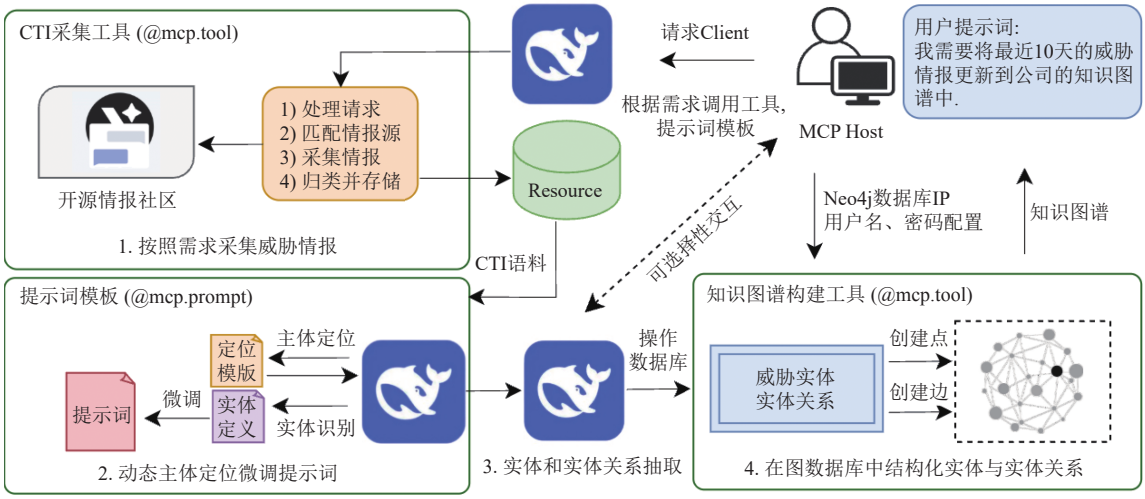


图 2 OPFA 框架整体设计图

3.1 请求与采集

传统的信息采集过程多为以数据库为基准的固定化查询, 用户往往难以通过模糊搜索精准获取到他们想要的信息. 而 MCP 服务器与大语言模型的结合解决了该问题. 大语言模型会通过语义感知用户的需求, 再把这些需求转化为 MCP 工具的输入参数. 工具会通过代码逻辑处理请求, 以匹配相关的情报源. 采集工作的目标主要是开源情报社区中的纯文本 CTI, 包括一些 APT 组织的攻击活动信息、漏洞信息以及恶意样本分析报告等. 工具会为这些信息做精细化分类. 这里, 语义需求到工具的参数转化是设计过程中的关键. 图 3 展示了一个具体的转化过程. 可以发现用户的需求可从语义上转化为 3 个参数的输入. 实现该工具的 MCP 服务器已经上传至在线平台: https://smithery.ai/server/@xue20010808/threatnews.

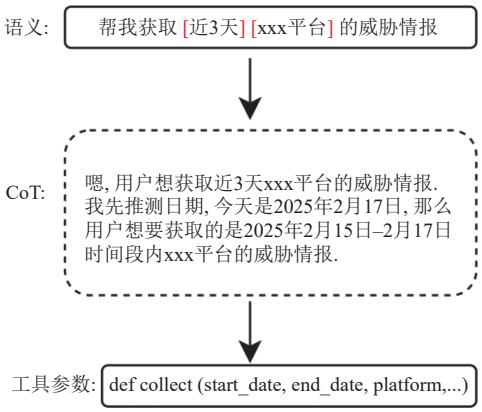


图 3 Semantic2Args (语义需求到工具参数的转化)

3.2 提示词微调

本文设计了一种新的方法来驱动大语言模型完成实体识别和实体关系抽取. 我们发现使用预设的提示词无法完全地定义所有 CTI 中出现的实体, 特别是在情报上下文过长的情况. 图 4 描述了这种现象. 这主要由于模型无法捕捉较远语义主体之间的联系.

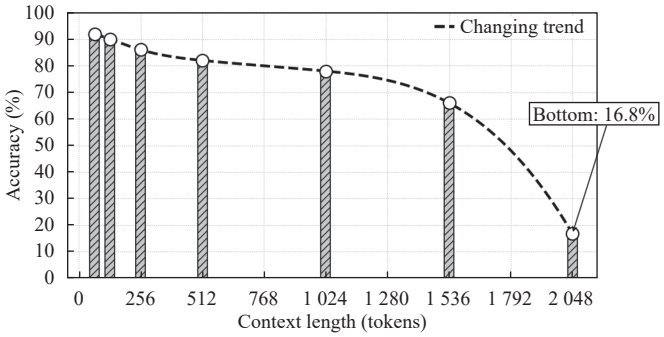


图 4 不同上下文长度下的实体识别精确率变化趋势

当上下文存在多个潜在实体且模型难以有效关联这些实体时, 目标实体的判定和抽取将面临显著挑战, 进而导致精确率下降. 我们使用动态语义主体定位技术改善了该问题. 设获取到的语料库为 $D=\{d_1,d_2,\ldots,d_N\}$, 其中 d_i 表示第 i 个 CTI 语料, 文档长度为 $L_i=|d_i|$, 主体集合 ε 为语料中出现的所有名词集合, e_k 为集合中的一个词元. 定义大语言模型输出的定位模板为 τ_d:

$$\tau_d=\left\{(e_k,p_k)\mid e_k\in\varepsilon,p_k=\frac{pos(e_k)}{L_i}\right\} \tag{1}$$

其中, p_k 为归一化相对位置, pos 用于计算主体在语料中的绝对位置. 聚合所有语料的定位模板生成全局覆盖的并集模板:

$$\tau^* = \bigcup_{d \in D} \tau_d \tag{2}$$

进一步细化其中的主体位置分布:

$$\forall e \in \varepsilon,\ P_e = \{p \mid (e,p) \in \tau^*\} \tag{3}$$

则并集模板可以表示为:

$$\tau^* = \{(e,P_e) \mid P_e \neq \varnothing\} \tag{4}$$

当新的语料加入 MCP 资源中, 则更新定位模板:

$$\tau^* \leftarrow \tau^* \cup \tau_{\text{new}} \tag{5}$$

$$\forall e \in \tau_{\text{new}},\ P_e \leftarrow P_e \cup \{P_{\text{new}}\} \tag{6}$$

那么, 只要动态地维护定位模板, 就能确保其准确识别出语料库中所有的 CTI 主体. 接着只需要驱动大语言模型将这些主体识别为对应的威胁情报实体 (不是所有的主体都是任务所需要的威胁实体) 并且输出为标准的实体定义即可. 模型会自动地将实体之间的关系进行交叉标注, 比如, 4 个实体之间通过排列组合可以建立 6 种关系. 模型会基于预设的提示词评估这些关系的强度, 并筛选掉联系较弱的关系.

提示词生成模型会根据实体与实体关系定义生成一段初始提示词. 对于大语言模型来说, 提示词的质量直接决定了模型执行任务的表现. 值得注意的是, 不同的大语言模型对于提示词的响应都会有自己的特点. 然而对于任务导向型的应用场景来说, 这便成为大语言模型的一种缺陷.

在本文中, 我们需要模型严格地按照定义识别实体与实体关系, 那么提示词中的一些语言细节便成为需要微调的目标. 该过程中, 需要微调的部分为除了定义以外的其他提示词, 如图 5 所示. 考虑到前向传播的稳定性, 我们使用本地模型作为输出模型. 对此, 本文设计了如下微调方法.

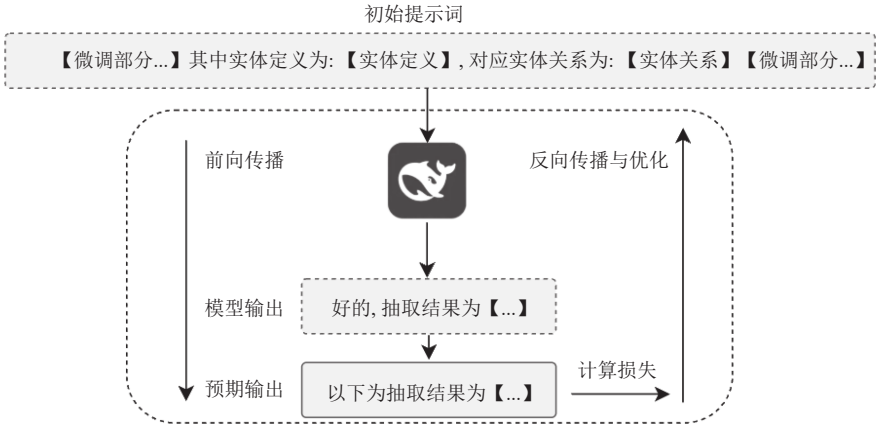


图 5 提示词微调过程

为了量化模型输出与预期结果的偏差, 引入余弦相似度和信息熵的联合优化目标. 定义模型输出集合为 E_{output}, 预期输出集合为 E_{target}, 两者的余弦相似度计算公式如下:

$$Sim_{\cos}(E_{\text{output}}, E_{\text{target}}) = \frac{\sum_{i}^{n} V_i^{\text{output}} \cdot V_i^{\text{target}}}{\left\| V_{\text{output}} \cdot V_{\text{target}} \right\|} \tag{7}$$

其中, V_i 为词向量, 通过预训练的词向量模型生成, 对应于大语言模型输出内容中的 token.

对于实体关系的抽取, 我们引入信息熵来衡量分布的合理性, 信息熵的计算公式如下:

$$H(R) = -\sum_{r \in R} P(r) \log P(r) \tag{8}$$

其中, R 为关系集合, $P(r)$ 为关系 r 在模型输出内容中出现的概率. 通过最小化 $H(R)$ 避免关系抽取的随机性. 最终微调目标函数如下:

$$T = \lambda(1 - Sim_{\cos}) + (1 - \lambda)H(R) \tag{9}$$

其中, $\lambda \in [0,1]$ 为平衡因子. 经过多轮微调, 我们可以得到一段最佳提示词模板, 由它驱动大语言模型进行实体和实体关系抽取.

3.3 知识图谱构建

OPFA 调用 MCP 服务器中的 Neo4j 数据库操作工具来自动化地完成知识图谱构建. 具体的工具名称、输入参数以及工具描述如表 1 所示.

表 1 Neo4j MCP 服务器工具

工具名称	输入参数	工具描述
Create_node	*entity*, *KG*	为每一个实体构建点
Create_relationship	*entity_relation*, *KG*	为每一个实体关系构建边
Execute_query	*query*	执行数据库查询

用户在客户端调用该 MCP 服务时, 需要提前配置好需要操控的目标 Neo4j 数据库信息. 之后客户端会根据模型输出顺序创建点与边, 从而在图数据库中构建一个个孤立的连通图. 接下来, 模型会调用数据库查询操作寻找是否有实体属性相似的节点, 如果有, 模型会继续调用边创建工具为这两个实体建立关联. 最终, 互相关联的威胁情报构成了完整的知识图谱. 算法 1 详细描述了该过程.

算法 1. 知识图谱构建工具调用算法.

输入: 提示词模板 *P*, 威胁情报 *CTI*;
输出: 知识图谱 *KG*.

1. 初始化: *KG*, *Model*
2. **for** *cti* ∈ *CTI* **do**
3. *entity*, *entity_relation* = *Model*(*P*, *cti*)
4. *Create_node*(*entity*, *KG*)
5. *Create_relationship*(*entity_relation*, *KG*)
6. **end for**
7. *Similarity_relation* = *Execute_query*(*query*, *KG*)
8. **if** *len*(*Similarity_relation*) > 0 **then**
9. **for** *similarity* **in** *Similarity_relation* **do**
10. *Create_relationship*(*similarity*, *KG*)
11. **end for**
12. **end if**

3.4 实体属性值扩充

考虑到那些只对实体信息进行粗略描述的威胁情报, 比如“攻击者使用 CVE-2024-50379 漏洞上传了 JSP 恶意文件, 成功获取了目标服务器的 Web 权限”, 当 OPFA 提取到实体 (漏洞), 并且为其在知识图谱中创建的实体属性值为 CVE-2024-50379 时, 这样的信息实际上对于用户来说意义不大. 它并不能帮助用户了解这个漏洞的背景信息, 这种情况下有必要对该属性值的内容进行扩充. 具体过程如算法 2 所示, 首先用户提出属性值扩充的需求, 想

要将图谱中所有关于漏洞的属性值进行扩充. 这时客户端大语言模型会按照第 3.1 节中所介绍的感知用户的语义, 再分析判断如何调用工具和资源. 这里, 如果 MCP 资源中存在对应漏洞的信息, 大语言模型会直接将该内容补充到 Neo4j 数据库中, 相当于实现了一次检索增强生成 (retrieval-augmented generation, RAG). 如果不存在, 模型则会调用采集工具到目标漏洞平台获取具体的漏洞细节.

算法 2. 实体属性值扩充算法.

输入: 扩充提示词 P_e, MCP 资源 R;
输出: 扩充后的 KG.

1. 初始化: *entity_new*
2. *Expand_value* = *Model*(P_e)
3. **if** *Expand_value* **in** *R* **then**
4. 模型调用工具更新对应 *entity* 的属性值
5. **else**
6. //收集目标属性信息
7. *value* = *Collect_CTI*(*Expand_value*, *URL*)
8. *entity_new*[*value*].*add*(*value*)
9. *Create_node*(*entity_new*, *KG*)
10. *relation* = *Relate*(*entity_new*, *entity*)
11. *Create_relationship*(*relation*, *KG*)
12. **end if**

如图 6 所示, 属性值扩充分为两种方式: 一种是直接在原实体中补充一条漏洞信息的属性值; 另外一种是额外创建一个实体, 作为原实体的解释实体.

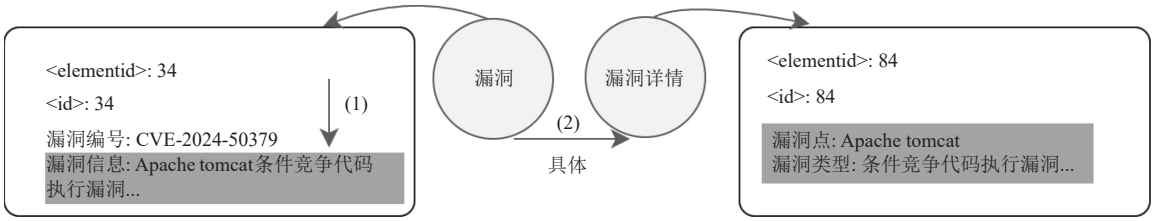


图 6 属性值扩充方式

4 实验分析

OPFA 采用 MCP 经典的 C/S 架构. 实验中, 我们在客户端选择了支持 MCP client 的 Cursor, 版本为 0.47.8, 并且使用 claude-3.7-sonnet 作为客户端模型. 服务端使用 3.11.7 版本的 Python 编写 CTI 采集工具以及 Neo4j 数据库操作工具, 均利用 FastMCP 库实现其与 MCP 服务端的通信. MCP 资源库由 Chroma 实现, 其提供对词向量的存储与读取功能. 提示词微调使用本地部署的 deepseek-r1:70B 作为输出反馈模型. 图形数据库使用版本号为 4.4.36 的 Neo4j. Host 侧使用的操作系统为 MacOS Sonoma 14.6.1.

我们首先对本文提出的动态主体定位技术进行实验验证. 接下来, 实验性地考量提示词的微调过程以验证其是否能够在 OPFA 运行过程中稳定收敛. 随后, 对比在使用大语言模型的基础上, 本文使用的动态实体定义方法与传统的预设定义方法, 评估它们在同一 CTI 集合中的抽取结果. 除此之外, 我们以人工标注为标准, 在精确率 (*Precision*)、召回率 (*Recall*) 和 *F*1 值 (*F*1-*Score*) 这 3 个维度对比大语言模型方法与深度学习方法的实体与实体

关系抽取能力. 最后, 通过一个 OPFA 的运行案例在 Neo4j 中展示知识图谱的构建结果, 并且分析其应用场景.

4.1 动态主体定位实验结果分析

为了验证动态主体定位技术在长上下文语料中的表现, 我们选取了 CTI 语料库 D 中 142 篇长度超过 2 048 个 tokens 的威胁情报作为大语言模型学习主体定位的训练集合. 本节将评估该方法训练出的定位模板在数据集中的主体定位效果如何. 本文设计了一个评估标准: 主体相似性 (subject similarity), 使用 Word2Vec 的 similarity 方法计算模板定位到的主体与标记主体词向量之间的余弦相似度. 图 7 展示了定位模板在 3 个不同长度语料中的定位表现. 可以发现, 相似度峰值处对应的 token 位置有 85% 都命中了真实 token 位置. 在实验中, 147 篇语料的平均命中率为 88.42%, 验证了该方法定位长文本主体位置的有效性.

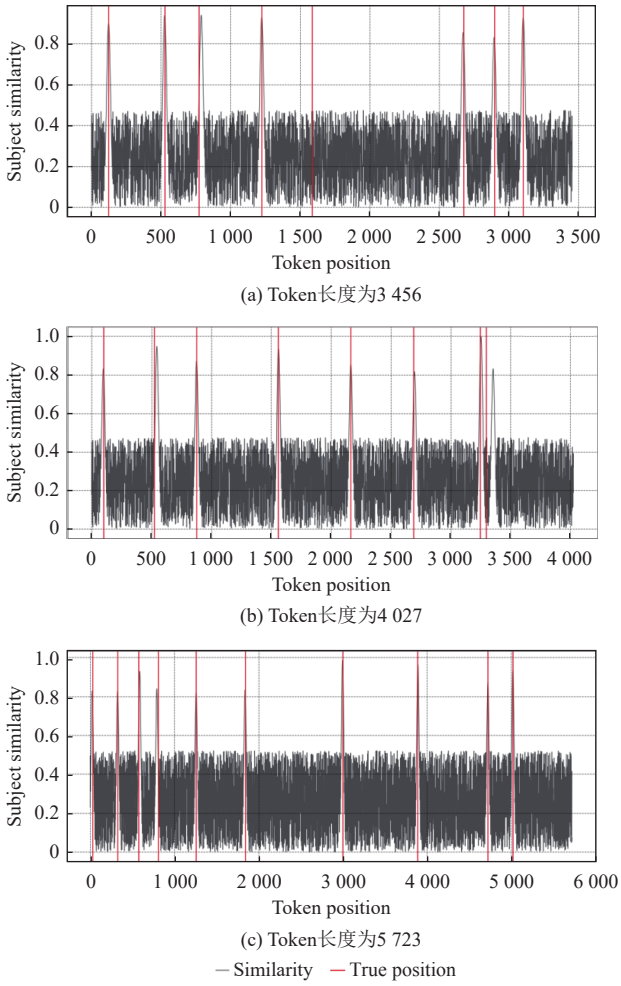


图 7 动态主体定位表现

定位模板最终在这 142 篇威胁情报中定位到了 71 个不同主体, 由大语言模型识别出 50 个存在的威胁实体, 交叉标注后总结出 64 种实体关系, 并对它们做出了定义.

4.2 提示词微调评估

在 MCP 的工作流中, 为了动态地满足大语言模型在处理特定任务中的即时性需求, 提示词微调是最高效的一种手段. 本文任务中, 由于 CTI 语料的频繁更新, 周期性地更新与微调提示词模板是必要的. 图 8 展示了一次微调记录, 即损失值及相似度随迭代次数的变化以及余弦相似度 (公式 (7) 计算所得) 的变化. 可以发现迭代 50 次后,

相似度收敛到了 0.88, 此时使用该提示词进行输入可以得到预期输出结果. 在 142 篇 CTI 语料加入 MCP 资源的过程中, OPFA 共微调了 12 次提示词模板, 平均相似度收敛可以达到 0.85.

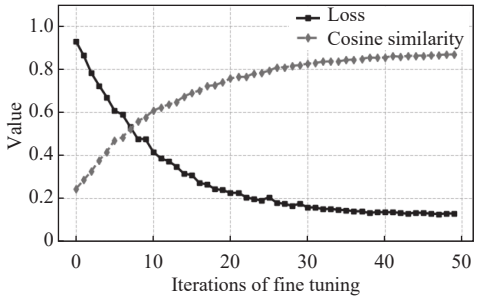


图 8　损失值及相似度随迭代次数的变化曲线

4.3　动态定义对比分析

本文所使用的大语言模型应用方法实际上是在动态过程中定义实体和实体关系的 (通过提示词驱动). 其他研究者在应用大模型实现 CTI 知识图谱构建的过程中采用了不同的方法. 预定义方法往往会利用启发式的定义尽可能地覆盖 CTI 中会出现的实体和实体关系. 表 2 展示了 3 种不同方法在相同 142 篇 CTI 语料中抽取的实体与实体关系结果. 实验中, 按照 LLM-TIKG[8]所使用的方法, 预定义了如下实体: 恶意软件、威胁类型、攻击者、技术手段、利用工具、利用漏洞、IP 地址、域名、URL、文件、哈希, 并且使用 Llama2-7B 模型进行指令微调. 按照 AutoCTI2KG[9]的方法, 将 ATT&CK 框架战术以及技术定义、CAPEC 攻击模式重写后直接输入模型.

表 2　对比预定义方法

方法类型	代表框架	形式	实体识别结果	关系抽取结果
预定义	LLM-TIKG[8]	典型定义+模型微调	40/52	51/66
	AutoCTI2KG[9]	ATT&CK+CAPEC	48/52	54/66
动态定义	OPFA	动态主体定位+提示词微调	50/52	64/66

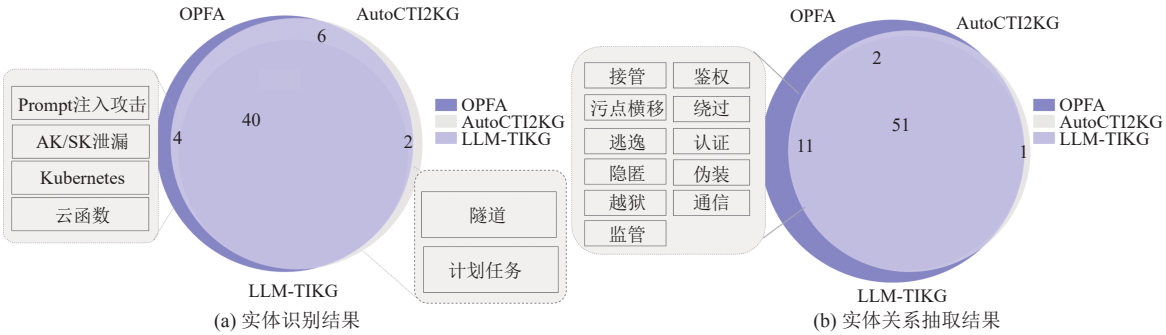
注: 51/66表示66个实体关系中成功抽取了其中的51个

本文实验中, 人工标注的实体共有 52 个, 实体关系有 66 个. 后文图 9 通过维恩图对比了 3 种方法在实体识别与关系抽取任务中的结果重叠与差异. LLM-TIKG 所使用的方法, 由于限定了实体定义的范围, 当 CTI 中出现与限定范围实体中关联度较低的实体时, 模型便无法对其进行识别. 而 AutoCTI2KG 利用既定的攻击模式框架来定义和分析 CTI, 在面对新兴威胁时存在滞后性. 在本文语料中, 其无法识别出包括 Prompt 注入攻击、AK/SK 泄漏、Kubernetes、云函数在内的新兴威胁实体, 因而也无法抽取和这些实体相关的实体关系, 例如接管、污点横移、逃逸、隐匿等. 同时, OPFA 在主体-实体识别过程中错失了一些有歧义性的实体, 例如隧道、计划任务等. 总体而言, 由于实体与实体关系的动态定义, OPFA 能够更好地抽取 CTI 语料中那些未出现过或还未被预先定义的实体与实体关系, 确保了 CTI 知识图谱对未知威胁和新型威胁的应急能力.

4.4　对比深度学习方法

OPFA 是基于大语言模型抽取方法的一次前瞻性扩展. 为了更全面地验证该方法的实效性, 我们抽取语料库中上下文长度小于 512 个 token 的 CTI (共 1 724 篇) 作为数据集, 在此基础上对比 OPFA 与传统深度学习实体与实体关系抽取方法. 同样地, 以人工标注为标准, 我们在精确率、召回率、$F1$ 值这 3 个维度上考量模型在相同威胁情报集合中的表现. 在实体识别与实体关系抽取任务中, 对于它们的定义如公式 (10) 所示. 其中, TP 是正确识别为实体/实体关系的数量, FP 是错误识别为实体/实体关系的数量, FN 为未被识别的实体/实体关系数量.

$$Precision=\frac{TP}{TP+FP},\ Recall=\frac{TP}{TP+FN},\ F1\text{-}Score=\frac{2TP}{2TP+FP+FN} \tag{10}$$



(a) 实体识别结果 (b) 实体关系抽取结果

图 9 OPFA、AutoCTI2KG 以及 LLM-TIKG 的实体关系抽取结果示意

表 3 展示了 OPFA 与其他 7 个基于深度学习模型的方法在实体识别任务中的表现对比. 本文方法在精确率、召回率和 $F1$ 值上均优于其他模型, 表明其在实体识别任务中具有显著性能优势. BiLSTM-CRF 和 BERT-base-NER 也表现出较好的性能, $F1$ 值达到了 0.8876 和 0.8347, 仅次于 OPFA.

表 3 实体识别表现对比

模型	精确率	召回率	$F1$值
BiLSTM-CRF[23]	0.9412	0.8312	0.8876
BERT-base-NER[2]	0.9091	0.7792	0.8347
DISCO Nets[31]	0.8750	0.7273	0.7962
AFE[32]	0.8333	0.6494	0.7295
CNN-GCN[33]	0.8852	0.7013	0.7987
Transformer[34]	0.7692	0.5195	0.6205
OPFA	0.9722	0.9091	0.9444
人工标注	1.0000	1.0000	1.0000

实体关系抽取表现对比如表 4 所示, OPFA 在精确率、召回率和 $F1$ 值上仍然优于其他模型, 这是由于其优秀的实体识别表现. 相比之下, CNN-GCN 和 Transformer 的表现较差, 表明它们在处理复杂关系时存在一定的局限性. BiLSTM-CRF 仍然保持着较好的性能, 说明传统混合深度学习模型依靠上下文建模和标签依赖性建模仍然能在一定程度上提高模型处理较长依赖文本语义的能力.

表 4 实体关系抽取表现对比

模型	精确率	召回率	$F1$值
BiLSTM-CRF[23]	0.9074	0.8315	0.8676
BERT-base-NER[2]	0.8814	0.7846	0.8304
DISCO Nets[31]	0.8522	0.7335	0.7891
AFE[32]	0.8250	0.6908	0.7551
CNN-GCN[33]	0.6667	0.4264	0.5143
Transformer[34]	0.5556	0.3194	0.4000
OPFA	0.9783	0.9552	0.9666
人工标注	1.0000	1.0000	1.0000

4.5 鲁棒性分析

我们从语料库中分层抽取上下文长度<512、介于 512–1 024 之间及>1 024 个 tokens 的 CTI 报告各 281 篇, 分别构成 CleanSet-I、CleanSet-II 与 CleanSet-III 基准测试集. 首先观察 OPFA 在基准测试集中进行实体识别和实体关系抽取的表现, 随后分别对每个测试集中的 CTI 个体引入 10% 的噪声数据 (拼写错误、词序乱序、无关字符插入), 15% 的模糊表述 (选中实体替换为同义泛指或者缩写), 5% 的上下文冲突 (在 CTI 文本中插入矛盾陈述), 再观

察 OPFA 在扰乱后测试集中的表现变化. OPFA 的鲁棒性测试结果如图 10 所示, 其中图 10(a)–(c) 是实体识别的测试结果, 图 10(e)–(f) 是实体关系抽取的测试结果, 实线代表未引入扰动前的表现, 虚线代表引入扰动后的表现.

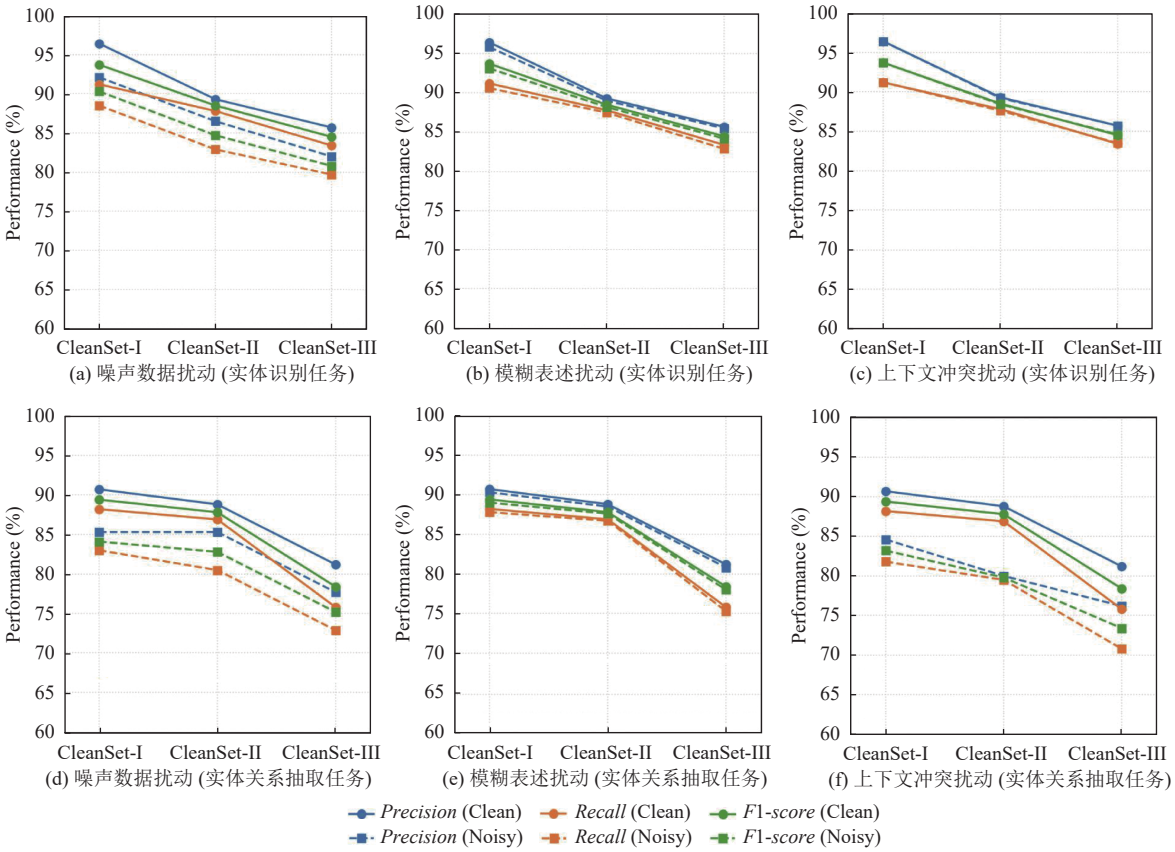


图 10 鲁棒性测试结果

图 10 的鲁棒性测试结果表明, OPFA 框架在面对多种数据扰动时, 整体上展现出良好的适应性与稳定性 (*F*1 值最大偏差小于 8%), 但其在不同任务、不同扰动类型下的表现存在差异性, 这与其底层技术机制密切相关.

在实体识别任务中, OPFA 对模糊表述扰动与上下文冲突扰动表现出较强鲁棒性. 如图 10(b) 与图 10(c) 所示, 其性能指标波动幅度控制在较低范围 (偏差普遍小于 2%). 这主要得益于大语言模型强大的多语义感知能力与本文提出的动态主体定位技术, 使得智能体能够穿透表述变化, 精准捕捉实体核心语义, 并对抗无关矛盾信息的干扰. 然而, 对于噪声数据扰动 (图 10(a)), 其召回率呈现相对明显的偏差. 分析认为, 这主要是由于那些严重错拼或乱码的字符序列已超出大语言模型的语义纠错边界, 导致其无法被识别为有效实体 (即假阴性 *FN* 增加), 但这部分噪声在真实数据中占比较低. 而对于词序变换等噪声, 其对实体识别的干扰甚微.

在实体关系抽取任务中, 其鲁棒性表现更为复杂. 如图 10(d) 所示, 噪声数据 (尤其是词序乱序) 对关系抽取的影响显著高于对实体识别的影响. 这主要是因为词序的混乱直接干扰了实体间的语法与语义依赖关系, 导致交叉标注过程失调, 即使实体本身已被正确识别, 它们之间的关系也难以被准确判定. 相较之下, 如图 10(e) 所示, 模糊表述对关系抽取的影响较小, 这从侧面印证了实体识别的高准确性是关系抽取可靠性的重要前提. 而对于上下文冲突扰动 (图 10(f)), 其影响机制则呈现出与文本长度的相关性: 在较短文本中, 矛盾语义与主体语义距离较近, 容易干扰智能体的关系判断; 随着 CTI 文本长度增加, 语义距离被拉大, 智能体在 MCP 架构的上下文管理下, 更能聚焦于主体语义关系而忽略局部矛盾, 因此扰动影响趋于减弱.

总体而言, 实验验证了 OPFA 框架在应对真实 CTI 数据中常见的不完美问题时, 具备可靠的鲁棒性基础, 尤其在长文本语境下的整体稳定性更为突出.

4.6 知识图谱构建结果展示

为了展示 OPFA 在真实使用场景下的 CTI 知识图谱构建能力, 本节将展示真实的构建过程与构建结果. 首先用户向 OPFA 提出构建需求: “采集最近 5 天的威胁情报, 并且构建相关知识图谱”. OPFA 接收到用户需求后连接至用户 Neo4j 数据库构建完整的知识图谱, 整个过程经历了 42 min 21 s, 具体节点与边个数随时间的变化曲线图如图 11 所示.

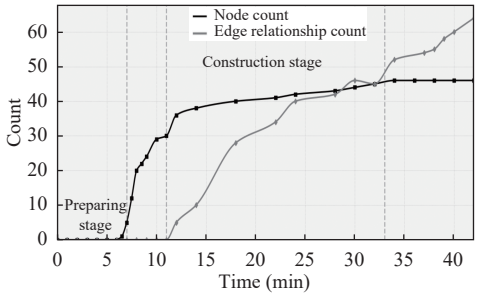


图 11 实体点和实体边随时间变化曲线

可以发现, OPFA 从开始运行到 6 min 这个阶段都在进行准备工作: 采集威胁情报信息、实体识别与实体关系抽取. 这之后, OPFA 开始在数据库中创建实体节点, 到 12 min 时已经创建了 36 个节点. 10 min 后, OPFA 开始同步创建实体关系边, 仅过了 14 min 就创建了 40 条边. 当任务结束时, OPFA 构建了一个包含 46 个实体点, 64 条实体关系边的 CTI 知识图谱.

Neo4j 中展示的节点与关系标签如后文图 12 所示. 实体点集合中出现了常见的攻击实体如钓鱼攻击、DDoS 攻击、SQL 注入等, 以及一些恶意文件, 包括常见的恶意软件、病毒、木马、勒索软件等. 实体关系集合中包含了威胁情报中经常会出现的动作, 如入侵、窃取、渗透、破解、隐藏等. 可以发现, OPFA 对于实体的定义与识别是细粒度的. 关于攻击者的实体定义被细致地划分为黑客组织、网络犯罪团伙、国家支持的攻击者、网络间谍等. 而受害者实体则被进一步地定义为教育机构、医疗保健机构、金融机构、关键基础设施、政府机构、企业、个人. 值得注意的是, 这些都是 OPFA 在构建过程中自主地从 CTI 中学习、定义与识别到的, 无任何人为干预. 细粒度定义使得知识图谱的更新和维护更加方便, 只需对相关实体和关系进行简单的调整和补充, 而不需要对整个知识图谱进行大规模修改. 不仅如此, 在威胁情报关联阶段, 细粒度定义使得 OPFA 能够更加精准地查询与匹配属性值相似的实体以建立关系.

由于该知识图谱过于庞大, 本文选择性地展示了部分威胁情报连通子图. 后文图 13 展示了与勒索软件 WannaCry 相关的知识图谱, 用户可以直观地了解到它的开发者、利用方式、传播方式以及哪些机构容易受其威胁. 点击其关联的恶意 IP 节点, 可以获取到与该软件通讯的 IP 地址以及其开放的端口号.

当用户查询到关于某个公司内部人员的相关威胁情报时, 如后文图 14 所示, 展示了受害用户 John Doe 从遭受社会工程学攻击, 到点击钓鱼邮件感染勒索病毒的溯源图. 公司可以学习这个威胁案例, 根据图谱中的攻击思路构建防守策略, 比如对员工进行防范钓鱼的安全意识培训, 封禁钓鱼平台域名所解析的 IP, 又或者按照知识图谱的提示模拟相关攻击进行攻防演练. 用户可以持续更新 CTI 知识图谱, 以及动态扩充相关实体的知识图谱, 单个提示词就可以执行所有操作.

4.7 扩展部署与用户反馈

第 4.6 节展示了 OPFA 在小范围场景下的真实构建结果. 本节将进一步介绍在企业内部封装及部署 OPFA 的具体细节, 并且收集企业安全情报运营人员的使用反馈以验证 OPFA 的实用性.

图 12 Neo4j 中创建的实体点与实体关系边

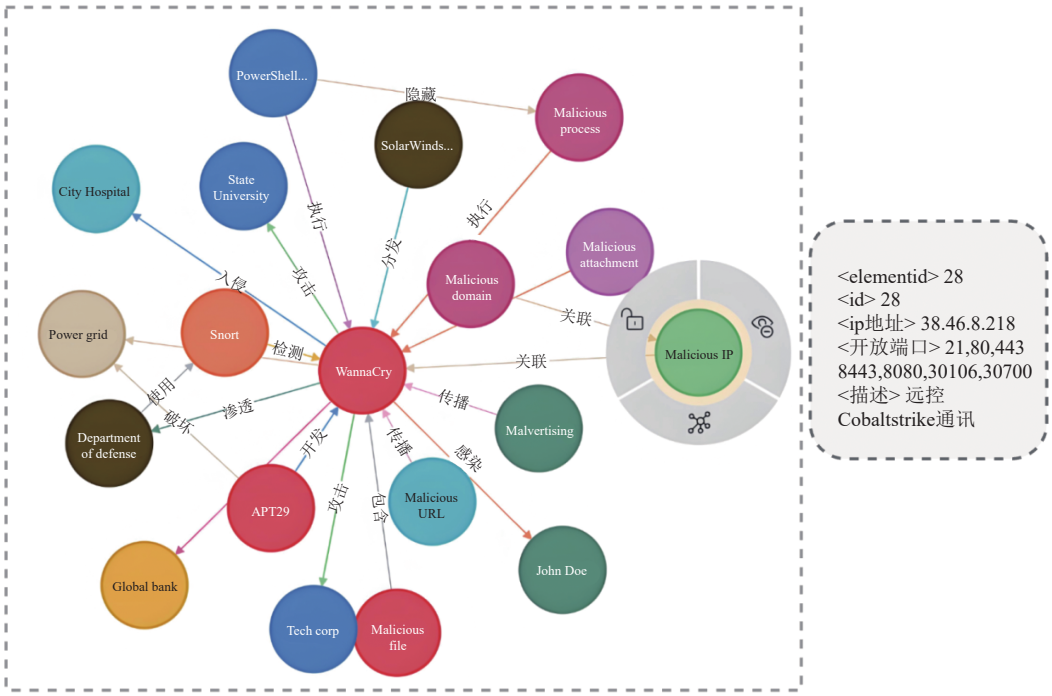


图 13 勒索软件威胁关联知识图谱

基于 OPFA 框架的整体设计, 其在企业内部的扩展部署方案严格遵循网络安全规范, 通过分区部署策略实现

核心数据不出网、外网访问可控的安全目标. 该方案将系统划分为核心内网区、数据交换区与外网区这 3 个逻辑部分, 确保各组件在受控环境中协同工作. 核心内网区部署 MCP Host 主智能体、Neo4j 图数据库及 Chroma 向量数据库, 所有知识图谱构建工具均严格限制于此区域, 禁止主动向外发起连接, 仅允许通过单向通信协议与数据交换区交互. 数据交换区作为缓冲地带, 部署大语言模型服务及缓存队列组件, 可根据企业安全要求灵活选择方案: 若采用公有云 API 调用, 则通过安全代理网关加密访问外部大语言模型服务; 若需更高安全性, 则直接部署本地化开源大语言模型, 确保所有数据处理流程均在内网完成. 外网访问权限仅限 CTI 采集工具, 通过防火墙策略严格限制其仅能从指定的开源威胁情报平台采集数据, 所有采集结果经数据交换区暂存后传输至核心内网区进行后续处理.

为实现高可用性与扩展性, 所有核心组件均采用微服务化与容器化部署, 通过负载均衡器分散请求压力, 避免单点故障. Neo4j 与 Chroma 数据库部署主从集群或分片集群, 保障数据存储的可靠性与查询性能. 在实际部署流程中, 用户通过自然界面向核心内网区的 MCP Host 提交任务, 驱动采集工具经安全通道获取外网情报, 随后调用数据交换区大语言模型服务进行语义分析与实体关系抽取, 最终由知识图谱构建工具将结构化数据写入内网图数据库. 该方案不仅可以保障企业内部威胁情报数据的安全性与隐私性, 还通过模块化架构支持弹性扩容, 以满足企业级应用对高性能与高可靠性的严苛需求.

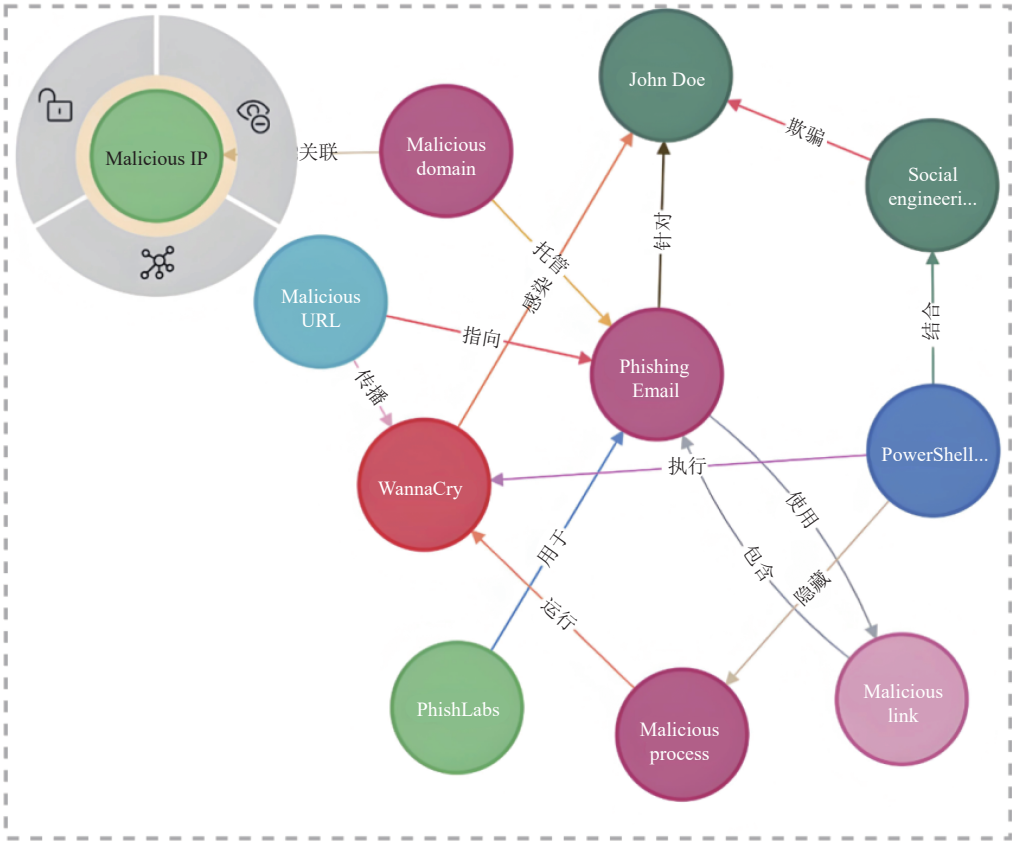


图 14　受害用户 John Doe 关联攻击溯源图

我们选取了安全情报运营部门过往的 34 个 CTI 知识图谱搭建需求, 并请该部门工作人员编辑对应的 OPFA 提示词以驱动智能体完成相同的任务. 接着, 我们记录了每个构建需求完成的时间周期, 并与过往人工完成的周期进行了对比, 结果如图 15 所示.

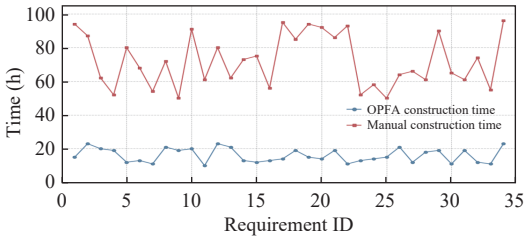


图 15　知识图谱构建周期对比

可以发现 OPFA 大大缩短了构建周期, 平均时间小于 24 h, 并且对于不同需求的处理时间是相对稳定的. 而手工搭建图谱的周期普遍大于 48 h, 且由于各种影响因素导致交付时间周期变化较大.

为了获得企业安全情报运营人员对于使用 OPFA 构建 CTI 知识图谱的真实反馈, 我们邀请了 9 名参与者填写调查问卷, 结果如表 5 所示.

表 5　问卷调查表

编号	评价内容	平均得分
Q1	构建过程几乎不需要人工介入	2.77
Q2	构建出的图谱质量较高	2.55
Q3	大大缩短了构建周期	3.00
Q4	如果需求清晰, 使用门槛很低	3.00
Q5	构建过程很好地遵循了安全合规和隐私保护需求	2.33

注: 评分0: 不认同; 1: 一般认同; 3: 非常认同

问卷反馈整体积极, 尤其“大大缩短了构建周期”和“如果需求清晰, 使用门槛很低”这 2 项获满分评价, 显示出 OPFA 在提升效率与易用性上的显著优势. “构建过程几乎不需要人工介入”得分接近满分, 表明 OPFA 的自动化能力已得到多数参与者认可. “构建出的图谱质量较高”亦获较高评分, 初步验证 OPFA 构建的 CTI 图谱有很好的可用性与可靠性. 此外, 安全合规项虽略低于其他项, 但仍处于中上水平, 反映出安全情报运营人员对 OPFA 在企业内部扩展部署的信任基础已初步建立.

5　总结与展望

针对网络空间威胁情报知识图谱构建中的效率与准确性难题, 本文提出了一种基于 MCP 智能体的通用威胁情报知识图谱构建框架 OPFA. 通过动态语义主体定位、提示词微调以及 MCP 资源调用等技术, 实现了从威胁情报采集到知识图谱自动化构建与扩充的全流程一体化. 实验结果表明, OPFA 框架在实体与实体关系抽取的精确率、召回率以及 $F1$ 值等方面均优于传统方法, 显著提升了知识图谱的构建效率和知识健壮性. 其构建的 CTI 知识图谱为威胁情报的深度利用奠定了坚实基础, 它能够显著辅助安全研究人员高效完成威胁研判、攻击溯源等安全分析任务, 在提升工作效率的同时, 也有助于沉淀和传承实战经验, 强化团队的整体安全知识水平. 进而, 基于图谱的关联分析能力, 可以更精准地识别潜在攻击链条与脆弱点, 为完善企业的事前预警与预防措施提供关键决策支持. 最终, 这对于构建企业的主动防御体系, 全面提升威胁感知敏锐度与应急响应能力, 具有十分重要的现实意义.

未来, 随着网络攻击手段的不断演变和威胁情报数据的日益复杂, 我们将进一步优化 OPFA 框架, 以适应更大规模、更复杂类型的情报数据处理需求. 同时, 探索如何将该框架与主动学习、强化学习等技术相结合, 实现知识图谱的自我进化与优化, 为网络安全防护提供更为精准、高效的情报支持. 基于此, 我们将继续研究多智能体协同框架的设计与优化, 聚焦于如何增强智能体之间的交互效率、任务分配机制以及联合决策能力, 从而提升整体系统的协作水平与复杂问题解决能力. 此外, 为应对不同应用场景下的多样化任务需求, 我们将系统性地开发针对性的语言模型微调技术. 重点将围绕领域适配、任务特性与资源约束这 3 个维度, 探索参数高效、数据驱动与提示优化相结合的高效微调方法. 通过引入领域知识增强、多任务联合学习及动态推理机制, 提升模型对专业场景的

语义理解与任务执行精度. 我们还将深化 OPFA 框架在医疗、金融等垂直领域的通用性与迁移潜力. 由于该框架基于 MCP 协议“资源-工具-提示”的松耦合架构设计, 使其具备天然的领域适应性. 通过替换 MCP 资源库中的领域文档、调整数据采集工具, 并利用提示词微调机制驱动大语言模型动态识别领域专属实体与关系, 可在最小化底层逻辑调整的前提下, 实现高效的领域迁移与适配.

References

[1] Wagner TD, Mahbub K, Palomar E, Abdallah AE. Cyber threat intelligence sharing: Survey and research directions. Computers & Security, 2019, 87: 101589. [doi: 10.1016/j.cose.2019.101589]

[2] Sun N, Ding M, Jiang JJ, Xu WK, Mo XX, Tai YH, Zhang J. Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. IEEE Communications Surveys & Tutorials, 2023, 25(3): 1748–1774. [doi: 10.1109/COMST.2023.3273282]

[3] Liu K, Wang F, Ding ZY, Liang S, Yu ZF, Zhou Y. Recent progress of using knowledge graph for cybersecurity. Electronics, 2022, 11(15): 2287. [doi: 10.3390/electronics11152287]

[4] Shi HY, Wei JX, Cai XY, Wang H, Gao SX, Zhang YQ. Research on threat intelligence extraction and knowledge graph construction technology. Journal of Xidian University, 2023, 50(4): 65–75 (in Chinese with English abstract). [doi: 10.19665/j.issn1001-2400.2023.04.007]

[5] Piplai A, Mittal S, Joshi A, Finin T, Holt J, Zak R. Creating cybersecurity knowledge graphs from malware after action reports. IEEE Access, 2020, 8: 211691–211703. [doi: 10.1109/ACCESS.2020.3039234]

[6] Mouiche I, Saad S. Entity and relation extractions for threat intelligence knowledge graphs. Computers & Security, 2025, 148: 104120. [doi: 10.1016/j.cose.2024.104120]

[7] Priya MKV, Glory HA, Sriram VSS. Enhanced deep learning for IIoT threat intelligence: Revealing advanced persistent threat attack patterns. In: Proc. of the 14th Int’l Conf. on Applications and Techniques in Information Security. Tamil Nadu: Springer, 2024. 201–217. [doi: 10.1007/978-981-97-9743-1_15]

[8] Hu YL, Zou FT, Han JJ, Sun X, Wang YL. LLM-TIKG: Threat intelligence knowledge graph construction utilizing large language model. Computers & Security, 2024, 145: 103999. [doi: 10.1016/j.cose.2024.103999]

[9] Liu JH, Zhan JY. Constructing knowledge graph from cyber threat intelligence using large language model. In: Proc. of the 2023 IEEE Int’l Conf. on Big Data (BigData). Sorrento: IEEE, 2023. 516–521. [doi: 10.1109/BigData59044.2023.10386611]

[10] Hou XY, Zhao YJ, Wang SN, Wang HY. Model context protocol (MCP): Landscape, security threats, and future research directions. ACM Trans. on Software Engineering and Methodology, 2025. [doi: 10.1145/3796519]

[11] Singh A, Ehtesham A, Kumar S, *et al*. A survey of the model context protocol (MCP): Standardizing context to enhance large language models (LLMs). 2025. https://www.preprints.org/frontend/manuscript/b45407370ad06ed48b5ebc462c1d8a2c/download_pub

[12] Pajo P. Accelerating AI integration: Multi-order effects and sociotechnical implications of standardized AI-tool interoperability. 2025. https://www.researchgate.net/publication/390232494_Accelerating_AI_Integration_Multi-Order_Effects_and_Sociotechnical_Implications_of_Standardized_AI-Tool_Interoperability

[13] Zhu KL, Du HY, Hong ZC, Yang XC, Guo SY, Wang Z, Wang ZHL, Qian C, Tang XR, Ji H, You JX. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In: Proc. of the 63rd Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers). Vienna: ACL, 2025. 8580–8622. [doi: 10.18653/v1/2025.acl-long.421]

[14] Wang CT, Li MD, Sun MX, Wang J, Yang XB, Niu JH, He ZY, Zhang WS. Cross-language bilayer knowledge graph with large-scale medical facts. Ruan Jian Xue Bao/Journal of Software, 2025, 36(3): 1240–1253 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7173.htm [doi: 10.13328/j.cnki.jos.007173]

[15] Zhu YS, Zhang W, Wang XK, Li ZY, Chen MY, Yao Z, Chen H, Chen HJ. Bidirectional imitation distillation for efficient incremental pre-training of E-commerce social knowledge graphs. Ruan Jian Xue Bao/Journal of Software, 2025, 36(3): 1218–1239 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7170.htm [doi: 10.13328/j.cnki.jos.007170]

[16] Xia JL, Li YL, Ge FP. Survey of entity relation extraction based on large language models. Journal of Frontiers of Computer Science and Technology, 2025, 19(7): 1681–1698. [doi: 10.3778/j.issn.1673-9418.2409086]

[17] Yang PA, Wu Y, Su LY, Liu BX. Overview of threat intelligence sharing technologies in cyberspace. Computer Science, 2018, 45(6): 9–18, 26 (in Chinese with English abstract). [doi: 10.11896/j.issn.1002-137X.2018.06.002]

[18] Li ZX, Li YJ, Liu YW, Liu C, Zhou NX. K-CTIAA: Automatic analysis of cyber threat intelligence based on a knowledge graph. Symmetry, 2023, 15(2): 337. [doi: 10.3390/sym15020337]

[19] Ren YT, Xiao YJ, Zhou YH, Zhang ZY, Tian ZH. CSKG4APT: A cybersecurity knowledge graph for advanced persistent threat organization attribution. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(6): 5695–5709. [doi: 10.1109/TKDE.2022.3175719]

[20] Li CJ, Yu H, Chen K, Zhao XJ, Han Y, Li AP. A survey of IoT threat intelligence knowledge graph. Journal of Cybersecurity, 2024, 2(2): 18–35 (in Chinese with English abstract). [doi: 10.20172/j.issn.2097-3136.240202]

[21] Böge E, Ertan MB, Alptekin H, Çetin O. Unveiling cyber threat actors: A hybrid deep learning approach for behavior-based attribution. Digital Threats: Research and Practice, 2025, 6(1): 2. [doi: 10.1145/3676284]

[22] Sarhan I, Spruit M. Open-CyKG: An open cyber threat intelligence knowledge graph. Knowledge-based Systems, 2021, 233: 107524. [doi: 10.1016/j.knosys.2021.107524]

[23] Huang ZY, Yu YN, Lin RM, Huang X, Zhang FL. Knowledge graph construction for network security base on modified BiLSTM-CRF. Modern Electronics Technique, 2024, 47(6): 15–21 (in Chinese with English abstract). [doi: 10.16652/j.issn.1004-373x.2024.06.003]

[24] Xiang XY, Gu ZQ, Zeng LY. A survey of cyber threat intelligence sharing and fusion technologies. Journal of Cybersecurity, 2024, 2(2): 2–17 (in Chinese with English abstract). [doi: 10.20172/j.issn.2097-3136.240201]

[25] Li ZY, Zeng J, Chen Y, Liang ZK. AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports. In: Proc. of the 27th European Symposium on Research in Computer Security on Computer Security (ESORICS 2022). Copenhagen: Springer, 2022: 589–609. [doi: 10.1007/978-3-031-17140-6_29]

[26] Ferrag MA, Ndhlovu M, Tihanyi N, Cordeiro LC, Debbah M, Lestable T, Thandi NS. Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices. IEEE Access, 2024, 12: 23733–23750. [doi: 10.1109/ACCESS.2024.3363469]

[27] Lai QN, Jin JD, Zhou CL. Research on knowledge graph construction technology for cyber threat intelligence based on large language models. Journal on Communications, 2024, 45(S2): 33–43.

[28] Zhang SQ, Li SH, Chen P, Wang SJ, Zhao CX. Generating network security defense strategy based on cyber threat intelligence knowledge graph. In: Proc. of the 1st Int'l Conf. on Emerging Networking Architecture and Technologies. Shenzhen: Springer, 2022: 507–519. [doi: 10.1007/978-981-19-9697-9_41]

[29] Meijer E. From function frustrations to framework flexibility: Fixing tool calls with indirection. Queue, 2025, 23(1): 19–38.

[30] South T, Marro S, Hardjono T, Mahari R, Whitney CD, Greenwood D, Chan A, Pentland A. Authenticated Delegation and Authorized AI Agents. arXiv:2501.09674, 2025.

[31] Bouchacourt D, Mudigonda P K, Nowozin S. DISCO Nets: Dissimilarity Coefficient networks. Advances in Neural Information Processing Systems, 2016, 29.

[32] Yang Y, Li I, Sang NJ, Liu LP, Tang XR, Tian QY. Research on Large Scene Adaptive Feature Extraction Based on Deep Learning. In: Proc. of the 7th Int'l Conf. on Computer Information Science and Artificial Intelligence. Shaoxing: ACM, 2024: 678–683.

[33] Alzubaidi L, Zhang J, Humaidi AJ, *et al*. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 2021, 8(1): 53.

[34] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez N, Kaiser L, Polosukhin I. Attention is all you need. In: Proc. of the 31st Conf. on Neural Information Processing Systems (NIPS 2017). Long Beach: Curran Associates Inc., 2017.

附中文参考文献

[4] 史慧洋, 魏靖烜, 蔡兴业, 王鹤, 高随祥, 张玉清. 威胁情报提取与知识图谱构建技术研究. 西安电子科技大学学报, 2023, 50(4): 65–75. [doi: 10.19665/j.issn1001-2400.2023.04.007]

[14] 王楚童, 李明达, 孙孟轩, 王静, 杨雪冰, 牛景昊, 贺志阳, 张文生. 融合大规模医学事实的跨语言双层知识图谱. 软件学报, 2025, 36(3): 1240–1253. http://www.jos.org.cn/1000-9825/7173.htm [doi: 10.13328/j.cnki.jos.007173]

[15] 朱渝珊, 张文, 王晓珂, 李志宇, 陈名杨, 姚祯, 陈辉, 陈华钧. 面向电子商务社交知识图谱高效增量预训练的双向模仿蒸馏. 软件学报, 2025, 36(3): 1218–1239. http://www.jos.org.cn/1000-9825/7170.htm [doi: 10.13328/j.cnki.jos.007170]

[16] 夏江镧, 李艳玲, 葛凤培. 基于大语言模型的实体关系抽取综述. 计算机科学与探索, 2025, 19(7): 1681–1698. [doi: 10.3778/j.issn.1673-9418.2409086]

[17] 杨沛安, 武杨, 苏莉娅, 刘宝旭. 网络空间威胁情报共享技术综述. 计算机科学, 2018, 45(6): 9–18, 26. [doi: 10.11896/j.issn.1002-137X.2018.06.002]

[20] 李昌建, 于晗, 陈恺, 赵晓娟, 韩跃, 李爱平. 物联网威胁情报知识图谱综述. 网络空间安全科学学报, 2024, 2(2): 18–35. [doi: 10.20172/j.issn.2097-3136.240202]

[23] 黄智勇, 余雅宁, 林仁明, 黄鑫, 张凤荔. 基于改进 BiLSTM-CRF 模型的网络安全知识图谱构建. 现代电子技术, 2024, 47(6): 15–21.

[doi: 10.16652/j.issn.1004-373x.2024.06.003]

[24] 向夏雨, 顾钊铨, 曾丽仪. 网络威胁情报共享与融合技术综述. 网络空间安全科学学报, 2024, 2(2): 2–17. [doi: 10.20172/j.issn.2097-3136.240201]

[27] 赖清楠, 金建栋, 周昌令. 基于大语言模型的网络威胁情报知识图谱构建技术研究. 通信学报, 2024, 45(S2): 33–43.

作者简介

沙乐天, 博士, 教授, CCF 专业会员, 主要研究领域为软件安全, 网络安全, 物联网安全.

薛磊, 硕士生, 主要研究领域为网络安全, 知识图谱, 大模型应用.

陈霄, 博士, 主要研究领域为系统安全, 虚拟化安全, 网络安全.

李德强, 博士, 讲师, 主要研究领域为智能系统安全, 对抗机器学习, 恶意软件检测.

肖甫, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为传感网, 物联网.