基于大模型与强化学习的网络威胁狩猎方法

崔泽源[1]，葛文翰[2]，王俊峰[1,2]
（1.四川大学视觉合成图形图像技术国防重点学科实验室，四川 成都 610065；
2.四川大学计算机学院，四川 成都 610065）

摘　要：网络威胁狩猎(CTH)通过主动发现攻击线索与恶意证据实现对攻击事件的快速响应。现有网络威胁狩猎方法虽具备在广泛信息源条件下进行决策的能力，但在现实场景中存在先验知识缺乏以及反馈稀疏的问题。针对上述问题，提出一种基于大语言模型(LLM)和强化学习(RL)的网络威胁狩猎算法 RE-HUNTER。为解决先验知识缺乏的问题，该方法构建了上下文向量数据库，利用大语言模型中的领域知识和网络威胁情报中的非结构化知识提升决策方法的冷启动效果，初始化强化学习权重；为解决反馈稀疏的问题，该方法改进了蒙特卡洛树搜索(MCTS)算法，引入了递归更新机制和方法相似度机制，从而增强了对实体和方法执行结果的反馈。基于 186 个真实攻击案例进行的网络威胁狩猎实验结果显示，该方法相较当前最优基线方法显著提高了搜索效率，在 0～2 000 步区间内平均召回率相对提升 18.24%；值得注意的是，在 0～250 步冷启动区间内较基线方法实现了 86.28%的平均召回率提升；此外，消融实验表明该方法的不同组成部分对实验结果均起到正向作用，能够有效降低网络威胁狩猎的成本。

关键词：网络威胁狩猎；主动防御；强化学习；大语言模型；蒙特卡洛树搜索
中图分类号：TP393.08　　**文献标志码：**A　　**DOI：**10.19678/j.issn.1000-3428.0070706

Cyber Threat Hunting Method Based on Large Language Model and Reinforcement Learning

CUI Zeyuan[1], GE Wenhan[2], WANG Junfeng[1,2]
(1.National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu 610065, Sichuan, China;
2.College of Computer Science, Sichuan University, Chengdu 610065, Sichuan, China)

【Abstract】Cyber Threat Hunting (CTH) enables rapid response to attack events through proactive discovery of attack clues and malicious evidence. Although existing CTH algorithms can search extensive information sources, they face challenges in real-world scenarios owing to insufficient prior knowledge and sparse feedback. To address these problems, this paper proposes a CTH algorithm, RE-HUNTER, based on Large Language Model (LLM) and Reinforcement Learning (RL). To address the lack of prior knowledge, this algorithm constructs a contextual vector database and leverages domain expertise from LLM and unstructured knowledge from cyber threat intelligence for cold-start decision-making to initialize RL weights. To address sparse feedback, this algorithm improves the Monte Carlo Tree Search (MCTS) algorithm by introducing a recursive update mechanism and method similarity to enhance the feedback on the execution results of both entities and methods. Experiments conducted on 186 real-world attack cases demonstrate that this model significantly improves the search efficiency compared with the current state-of-the-art baseline methods. Within the 0-2 000 step range, the average recall rate achieves a relative improvement of 18.24%. Notably, in the 0-250 step cold-start scenario, the average recall rate achieves a relative improvement of 86.28% compared to that achieved by the best baseline method. Furthermore, the ablation experiments indicate that each component of the proposed algorithm positively contributes to the overall performance, effectively reducing the cost of CTH.

【Key words】Cyber Threat Hunting (CTH); proactive defense; Reinforcement Learning (RL); Large Language Model (LLM); Monte Carlo Tree Search (MCTS)

0　引言

在当前网络技术发展、人工智能活跃的时代背景下，全球网络攻击呈现多样化、复杂化以及组织化的态势，攻击者在攻击方式、攻击工具及隐蔽性方面不断演进。传统基于知识库更新、关联规则匹配的

基金项目：国家自然科学基金(U24B20147，U2133208)；四川省重点研发计划(2024ZHCG0195，2024ZDZX0044，2024ZYD0269)。
作者简介：崔泽源，男，硕士研究生，主研方向为网络与信息安全；葛文翰，博士研究生；王俊峰(通信作者)，教授、博士。
收稿日期：2024-12-16　　**修回日期：**2025-03-20　**E-mail：**wangjf@scu.edu.cn

计算机工程 www.ecice06.com

被动防御方式,难以在第一时间识别出潜在的、不断变化的威胁,给网络安全防护带来了严峻挑战,并对相关企业和组织造成了巨大的经济损失[1]。在此背景下,网络威胁狩猎(CTH)作为一种主动防御策略,通过不断发现潜在的威胁线索和新出现的恶意证据,以期在实际攻击造成危害之前,捕获安全威胁,实现对攻击事件的及时响应和处置[2-3]。

近年来,以强化学习(RL)和大语言模型(LLM)为主的机器学习方法在网络安全领域展现出巨大的潜力[4-5]。强化学习通过与环境的交互,能够自主学习最优策略,已被应用于入侵检测、恶意流量识别等领域。大语言模型则具备对海量非结构化文本数据的理解生成能力,已被用于提取和分析网络威胁情报[6-7]。虽然已有网络威胁处理及网络威胁狩猎的应用场景,但是现有工作尚未有效融合强化学习的在线学习能力与大语言模型的领域知识泛化优势,导致在有限时间、受限资源的条件下网络威胁狩猎的数据采集效率与准确度难以提升[8-10]。具体表现为:

1)先验知识缺乏:尽管现有方法使用历史经验优化搜索策略,但是当本地数据或先验知识不足时,网络威胁狩猎决策算法面临冷启动问题,导致其在初始化时决策效果较差。

2)反馈稀疏:在现实场景下进行网络威胁狩猎,多源异构威胁样本数据质量不同,导致产生反馈稀疏的问题,增加了网络威胁狩猎模型拟合难度。

鉴于此,本文提出了一种基于大语言模型和强化学习的网络威胁狩猎算法。该算法旨在解决现实场景中存在的先验知识缺乏以及反馈稀疏的问题,提高网络威胁狩猎的效率和准确性。本文的主要贡献如下:

1)提出了一种结合大语言模型和强化学习的网络威胁狩猎方法 RE-HUNTER,针对先验知识缺乏与反馈稀疏等现实问题进行了有效的解决。

2)设计了基于改进蒙特卡洛树搜索(MCTS)[11]的网络威胁狩猎方法,引入递归更新机制和方法相似度机制,增强了模型对稀疏反馈的适应性。

3)构造了基于大量真实攻击案例的实验数据集,在 186 个真实攻击事件中进行网络威胁狩猎实验,设计了包括平均召回率、达到特定召回率的平均步数和平均累计奖励在内的多维评估指标。通过与多种基线方法的对比实验和详细的消融实验,验证了 RE-HUNTER 方法的整体有效性,量化分析了各组成模块对方法性能的具体贡献。

1 相关工作

1.1 网络威胁狩猎

网络威胁狩猎是一种主动的威胁检测策略,其通过在网络中积极搜索潜在威胁暴露点,建立全局攻击视图的方法,探查网络威胁发展态势,减少网络入侵造成的影响[1]。

当前的主流研究方法依赖自然语言处理技术,从结构化和半结构化的不同数据源进行威胁检索、情报分享[12-14],这些数据源为网络威胁狩猎提供丰富且彼此交叉的检索渠道[15],帮助研究人员快速了解攻击发展态势与规模,制定防御反制策略。不同研究从审计日志[9-10]、场景知识[16]、威胁情报[3,17]等数据源进行网络威胁狩猎。

溯源图是一种被广泛用于网络威胁狩猎中威胁场景描述与组织的方法[16,18],其通过图结构反映攻击系统中主体(如软件进程、线程等)和客体(如文件、网络套接字等)之间的交互情况。溯源图能够定义攻击受灾范围和系统运行状态。例如,文献[3]提出的 Extractor 方法能够从威胁情报中提取溯源图,将数据源中非结构化信息抽取为基于妥协指标(IoC)实体与调用关系的结构化知识图谱,进而发现更多高价值目标与高层次知识,辅助本地重建攻击场景。然而,基于溯源图进行网络威胁狩猎的效果极大程度依赖于信息源本身的质量,其局限性包括高昂的经验采集和初始化成本、大规模数据集标注的复杂性,以及在数据融合与实时响应方面的不足[15]。

因此,当前网络威胁狩猎面临的首要挑战是如何减少对专家高质量经验的依赖性,降低实际应用中的使用门槛,并提升从不同数据源中挖掘威胁情报的能力和效率[12-13],特别是如何能够高度智能化地结合异构知识提供攻击的整体视图[19-21]。

面对此类直接的效率需求,部分新兴研究考虑利用不同数据源之间的连接性降低数据获取成本和难度。文献[2]提出的 HinCTI 方法在异构信息网络上进行网络威胁狩猎,以弥补信息的不完整性。文献[8]提出的 MABAT 方法将网络威胁狩猎决策作为强化学习过程,提高网络威胁狩猎决策及响应效率。上述研究成果大多基于实验室充足或模拟的数据环境开展实验,这使得在实际操作中,这些方法由于数据源组成不同、数据实时收集成本和狩猎响应速度等问题,难以真正迁移至开放、动态、对抗的真实狩猎环境中。因此,本文的核心出发点在于提升网络威胁狩猎工具的效率与狩猎效

计算机工程 www.ecice06.com

果、降低使用成本。

1.2 强化学习与大语言模型

基于高效率、低成本的出发点，网络威胁狩猎任务需要尽可能降低对专家处理流程的依赖，从而实现决策流程的优化。强化学习和大语言模型对多种任务的良好兼容性，使得这个目标成为可能。强化学习因其在复杂决策中的应用而受到广泛研究[5,11]，在不同网络安全下游任务中展示出较好效果[22-23]。对于图上强化决策问题，文献[24]将图上游走问题建模为马尔可夫决策过程，证明了在图上进行查询决策的可行性。MABAT 方法采用了一种多臂赌博机(MAB)的变体进行网络威胁狩猎决策[8]，通过单步决策实现对可靠数据源的选择。文献[25]提出的方法利用强化学习进行本体扩展，展示了强化学习技术在网络安全应用中的可行性。此外，ε-greedy 等经典策略也可用于网络威胁狩猎任务[26]。上述建模虽然可以实现威胁检测或信息采集过程的主动迭代，但还存在一些不足：其一是未考虑到真实世界任务动作的复杂性，这种复杂性体现为待决策的动作组成丰富、状态空间复杂，因此需考虑可以进行多重决策的强化学习建模方法；其二是难以量化的决策反馈。由于大多数网络威胁狩猎发生于攻防对抗的过程中，决策过程具有不确定性以及长效依赖性，需要强化学习模型具备中长期记忆的能力。

大语言模型通过从大量文本数据集中获取世界知识，在下游任务中具有较强的零样本学习能力[27]。已有研究表明，大语言模型可以在开放世界场景与网络安全的特定领域应用场景中增强决策能力[28-29]。特别地，不同研究说明了大语言模型在冷启动场景中表现出显著辅助作用[30-32]，能够缓解先验知识不足的问题。上下文信息的质量显著影响了大语言模型对任务的理解，从而影响决策结果。检索增强生成(RAG)技术可以通过融合外部数据库的知识，将任务上下文信息整合到大语言模型中[33]。在网络安全领域，大语言模型已被应用于代码生成[34]、流量分析[35]、异常检测[36]、技战术分类[37]等任务[6-7]，取得了较传统方法更好的效果。这些方法通过大语言模型将任务拆解为原始文本，借助大语言模型的模型能力与上下文记忆能力实现深度检索、逻辑推理和语义摘要。因此，本文通过采用大语言模型辅助强化学习决策的方式，在无需训练或仅需少量提示的前提下来指导决策任务的初始化。

2 基于大模型与强化学习的网络威胁狩猎方法

2.1 问题定义

本文将网络威胁狩猎建模为分层马尔可夫决策过程(HMDP)，不仅能够考虑当前的状态，还能在决策中形成动作依赖，可以通过状态转移将奖励反向传播至早期动作。具体形式化定义如下：

狩猎威胁在 t 时刻的状态空间表示为 $s_t=(N_t,E_t,C_t)$，其中，$N_t=\{\langle k,v\rangle\}$ 表示当前已发现的威胁实体集合，k 是威胁实体类型标签，v 是威胁实体的名称，代表潜在的安全威胁，E_t 表示实体关系集合，$C_t\subseteq N_t$ 表示当前可操作的待选实体子集。

动作空间形式化表示包含 2 个层次：节点选择层从待选实体集合 C_t 中选择一个待扩展实体 n_i；方法选择层从实体 n_i 支持的网络威胁狩猎方法集合 $M(n_i)$ 中选择一个方法 a_j。该过程可表示为 $A_{i,j}(s_t)=\{(n_i,a_j)\mid n_i\in C_t,a_j\in M(n_i)\}$。经过该次迭代，实体 n_i 将会经过扩线方法 a_j 获得扩展，以产生待选实体集合 $C_{i,j}$，并依照扩线方法 a_j，构建新的威胁实体关系集合 $E_{i,j}=\{\langle n_i,a_j,n_{i,j,k}\rangle\mid\forall n_{i,j,k}\in C_{i,j}\}$。

状态转移形式化表示为 $P(s_{t+1}\mid s_t,n_i,a_j)$，在选择 (n_i,a_j) 后，当前状态 s_t 被更新为 s_{t+1}。该更新过程主要包含 3 个步骤：

第 1 步为实体集合更新：$N_{t+1}=N_t\cup C_{i,j}$，即将当前选择的实体 n_i 的待选实体集合 $C_{i,j}$ 加入已发现的威胁实体集合 N_t 中。

第 2 步为关系集合更新：$E_{t+1}=E_t\cup E_{i,j}$，即新的实体关系集合 $E_{i,j}$ 将被整合入原有关系集合 E_t 中。

第 3 步为待选实体集合的更新：$C_{t+1}=(C_t-\{n_i\})\cup C_{i,j}$，即从当前可操作实体集合中首先移除已执行扩展的实体 n_i，并加入新发现的实体 $C_{i,j}$。

决策目标是最大化经过 t 步决策后实体集中的恶意节点子集，即 $\max|N_t\cap M|$，其中，N_t 为经过 t 步决策后累积的实体集合，M 表示恶意实体集合，可通过不同网络威胁狩猎数据源进行标记。

2.2 框架设计与描述

RE-HUNTER 的总体结构如图 1 所示，主要包括上下文准备模块和网络威胁狩猎环境两大部分。

1)上下文准备模块。该模块的功能是根据网络威胁事件，从不同来源的网络威胁情报中，挖掘相关实体、对应文本等，目的是为网络威胁狩猎决策过程提供初始威胁实体，以及为大语言模型冷启动模块

提供威胁实体上下文。

2)网络威胁狩猎环境。该环境由强化学习决策算法和基于大语言模型的冷启动模块组成。

(1)网络威胁狩猎决策模块。决策模块负责执行网络威胁狩猎决策算法的主要流程,包括实体选择、方法选择、执行网络威胁狩猎方法以及反馈等。通过改进蒙特卡洛树搜索算法,决策模块能够在先验知识缺乏以及反馈稀疏等条件下高效进行网络威胁狩猎。

(2)基于大语言模型的冷启动模块。该模块旨在初始化网络威胁狩猎环境,主要通过将大语言模型的领域知识和先验经验通过权重生成的方式注入网络威胁狩猎决策模块,实现了在缺乏先验知识条件下的快速迭代。

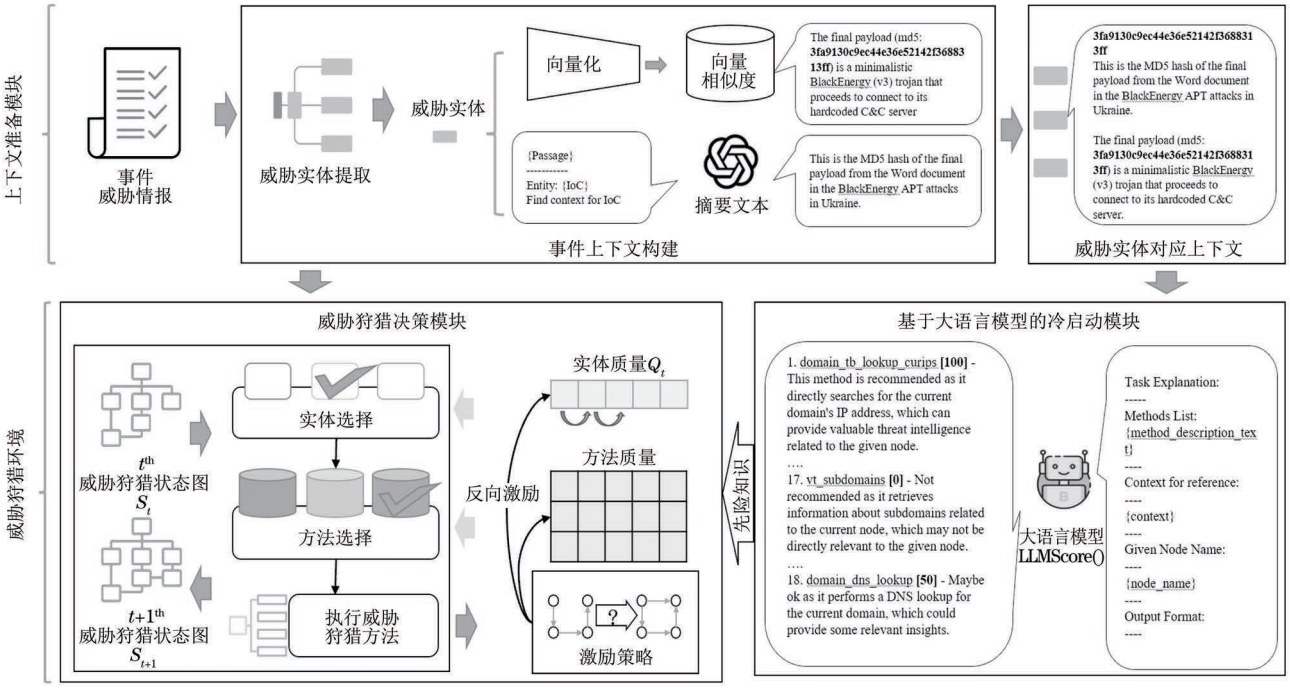


图 1 RE-HUNTER 的总体结构

Fig. 1 The overall structure of RE-HUNTER

2.3 上下文准备模块

对于目标事件的威胁情报,模块首先进行威胁实体提取,形成初始威胁实体集合,表示为 $N_D=\{\langle k,v\rangle\}^{|D|\times l}$。其中,$|D|$ 表示网络威胁情报初始集中的文件数量,l 是单个文件中威胁实体的最大数量。

为了提升大语言模型的决策能力,本文提出了一种联合上下文提取方法,构建两种类型的上下文:基于相似性的事件上下文和基于摘要的上下文。

2.3.1 基于相似性的事件上下文构建方法

基于相似性的事件上下文构建方法负责初步使用语义相似度度量来衡量待评估实体在语义空间中的位置,主要依赖于词嵌入模型。

首先,将威胁情报 T 按句分割为待处理文本片段 T'_i,然后使用 BERT 模型将其转换为高维空间中的向量 $\boldsymbol{V}_{\mathrm{BERT}i}$。每个向量 $\boldsymbol{V}_{\mathrm{BERT}i}$ 是其原始文本片段 T'_i 的编码映射结果,最终形成一个向量到原始文本的数据库 DB_{BERT},主要过程如式(1)所示:

$$\mathrm{DB}_{\mathrm{BERT}}=\{\langle \boldsymbol{V}_{\mathrm{BERT}i},T'_i\rangle \mid T'_i\in T,\boldsymbol{V}_{\mathrm{BERT}i}=\mathrm{BERT}(T'_i)\} \tag{1}$$

式中:BERT()是基于 Transformer 的双向编码器表示模型(BERT)的编码函数。

2.3.2 基于摘要的上下文匹配方法

对于哈希等类型的威胁实体,其名称并未隐含更多信息,因此有必要基于威胁情报进行相关摘要的上下文匹配与构造。对于每个 t 次扩展后威胁实体集合 N_t 中的实体 $n_i\in N_t$,生成摘要文本 $T_{\mathrm{LLM}i}$ 及其对应的向量 $\boldsymbol{V}_{\mathrm{LLM}i}$,并形成上下文数据库 $\mathrm{DB}_{\mathrm{LLM}}$。上下文数据库 $\mathrm{DB}_{\mathrm{LLM}}$ 的构建过程如式(2)所示:

$$\begin{aligned}&\mathrm{DB}_{\mathrm{LLM}}=\{\langle n_i,\boldsymbol{V}_{\mathrm{LLM}i},T_{\mathrm{LLM}i}\rangle \mid \forall n_i\in N_t\}\\&\text{s.t. } T_{\mathrm{LLM}i}=\mathrm{Context}(n_i,\mathrm{CTI}(n_i)),\\&\qquad \boldsymbol{V}_{\mathrm{LLM}i}=\mathrm{BERT}(T_{\mathrm{LLM}i})\end{aligned} \tag{2}$$

式中:Context 表示基于大语言模型和思维链的上下文摘要函数;$\mathrm{CTI}(n_i)$ 表示获得包含对应实体的威胁情报文本。

对于待查找的每个威胁实体，使用相似度函数 Sim 在 DB_{BERT} 数据库中搜索，匹配到与威胁实体向量 $\boldsymbol{V}_{IoC}$ 最相似的文本片段 T_{text} 及其对应的向量 $\boldsymbol{V}_{text}$，如式(3)所示：

$$(T_{text},\boldsymbol{V}_{text})=\underset{(T_i,\boldsymbol{V}_i)\in DB_{BERT}}{\operatorname{argmax}}\ \mathrm{Sim}(\boldsymbol{V}_{IoC},\boldsymbol{V}_i) \tag{3}$$

式中：Sim 函数使用向量余弦相似度进行实现。

此外，在DB_{LLM} 中，使用威胁实体名称作为关键词搜索相关的摘要上下文文本 T_{LLM} 及其对应的向量 $\boldsymbol{V}_{LLM}$，该过程可以表示为式(4)：

$$(T_{LLM},\boldsymbol{V}_{LLM})=\mathrm{Find}(n_i,DB_{LLM}) \tag{4}$$

式中：$\mathrm{Find}(n_i,DB_{LLM})$ 表示在DB_{LLM} 数据库中使用威胁实体作为关键词进行搜索的过程，使用向量余弦相似度求 Top-3 相似语句。上述两类决策文本都将作为与威胁实体相关的上下文输入大语言模型，帮助大语言模型实现冷启动决策优化。

2.4 网络威胁狩猎决策模块

本文的核心方法为网络威胁狩猎决策算法，这是一种基于环境反馈的强化学习方法，实现了分层马尔可夫决策过程，使网络威胁狩猎模块可以发现新威胁实体。网络威胁狩猎过程如算法 1 所示。每轮网络威胁狩猎开始时，先选择当前最优威胁实体，而后根据是否为冷启动状态选择最优网络威胁狩猎方法，扩展后进行反向传播激励。

算法 1 RE-HUNTER 决策算法

输入 实体的初始集合 N_0，最大迭代次数 T

输出 优化的网络威胁狩猎路径和决策结果

1. $N_t \leftarrow N_0$，$t \leftarrow 0$，$S_0=\phi$
2. 当 $t<T$ 时：//开始每轮网络威胁狩猎
3. $n_i \leftarrow \mathrm{select_node}(N_t)$
4. $A_i \leftarrow \mathrm{expand_actions}(n_i)$
5. 如果 $n_i \in N_0$，则：
6. $a_j \leftarrow \mathrm{select_action_coldstart}(A_i,n_i)$
7. 否则：
8. $a_j \leftarrow \mathrm{select_action}(A_i,n_i)$
9. $C_{i+1} \leftarrow \mathrm{execute_action}(n_i,a_j)$
10. $N_{t+1} \leftarrow N_t \cup C_{i+1}$
11. $S_{t+1} \leftarrow \mathrm{update_state}(S_t,n_i,a_j,C_{i+1})$
12. $\mathrm{update_node_rewards}(S_{t+1})$
13. $\mathrm{update_method_rewards}(S_{t+1})$
14. $t \leftarrow t+1$
15. 返回 S_T

2.4.1 实体选择过程

实体选择过程通过从待扩展实体集合中选择当前最优的威胁实体，定义 n_0 为初始网络威胁狩猎起点，$N_0=\{n_0\}$ 为初始实体集，$S_0=\phi$ 为初始状态。

实体选择过程首先使用 select_node 函数从时刻 t 的威胁实体集合 N_t 中选择一个当前最优威胁实体 $n_j=<k_i,v_i>$。式(5)所定义的函数 R 表示函数 select_node 对应的决策过程，该函数在给定时间 t 从集合 N_t 中选择当前最优的实体。

$$R(N_t)=\underset{n_i\in N_t}{\operatorname{argmax}}\left(\frac{Q(n_i)}{N(n_i)}+c\sqrt{\frac{\ln N(P(n_i))}{N(n_i)}}\right) \tag{5}$$

该式是置信上界树搜索算法(UCT)公式的变体，使用质量评估函数 Q 对威胁实体 $n_i\in N_t$ 进行评估打分。$R(N_t)$ 的其余项是传统 UCT 策略所需的参数，包括对 n_i 及其父节点 $P(n_i)$ 的访问次数的统计 $N(n_i)$ 和 $N(P(n_i))$，以及超参数 c，用于控制探索与利用权重。与上置信界算法(UCB)相比，UCT 通过质量评估函数引入了层级依赖，以体现攻击链的路径累积效应，而 UCB 仅考虑单步奖励。

2.4.2 方法选择过程

在选择出当前最优威胁实体 n_i 后，根据当前最优实体对应的最优网络威胁狩猎方法 a_j，首先，调用 expand_actions 函数求出对于当前最优实体 n_i 与对应当前实体类型 k_i 的网络威胁狩猎方法集合 A_i，排除在此实体中先前访问过的网络威胁狩猎方法。当前实体类型包括哈希、域名、URL 链接、路径等。而后，调用 select_action 函数根据先前的扩展经验 $\boldsymbol{W}_t$ 从 A_i 中选择最佳网络威胁狩猎方法 a_j。其中，$\boldsymbol{W}_t$ 是权重矩阵，如式(6)所示：

$$\boldsymbol{W}_t=\begin{bmatrix} w_{a(k_1)_1} & w_{a(k_1)_2} & \cdots & w_{a(k_1)_j} & \cdots \\ w_{a(k_2)_1} & w_{a(k_2)_2} & \cdots & w_{a(k_2)_j} & \cdots \\ \vdots & \vdots & & \vdots & \\ w_{a(k_i)_1} & w_{a(k_i)_2} & \cdots & w_{a(k_i)_j} & \cdots \\ \vdots & \vdots & & \vdots & \end{bmatrix} \tag{6}$$

在该矩阵中，不同威胁实体类型 k_i 的第 j 个信息扩展方法表示为 $a(k_i)_j$ 及其对应权重 $w_{a(k_i)_j}$。当方法集合 A 输入时，若当前 n_i 没有被选择过，则当前方法选择过程为冷启动模式，使用 select_action_by_coldstart 函数，根据 $W_{coldstart}(A)$ 实现最优网络威胁狩猎方法选择，如式(7)所示；否则使用 select_node 函数，根据 $W_{common}(A)$ 进行最优网络威胁狩猎方法选择，如式(8)所示。

$$W_{coldstart}(A)=\underset{a_j\in A}{\operatorname{argmax}}\left(\frac{w_{a_j}}{N(a_j)}+\mathrm{LLMScore}(a_j)\right) \tag{7}$$

www.ecocoo.com

$$W_{\text{common}}(A)=\underset{a_j\in A}{\operatorname{argmax}}\left(\frac{w_{a_j}}{N(a_j)}+c\sqrt{\frac{\ln N(A)}{N(a_j)}}\right) \tag{8}$$

式(7)解决了冷启动阶段的方法选择问题。在冷启动阶段,关于不同威胁方法有效性的先验信息较少,对于每个动作 $a_j\in A$,其有效性基于其固有权重值 w_{a_j} 与动作被执行的次数 $N(a_j)$归一化的加权组合,以及从大语言模型输出的额外得分 LLMScore(a_j)进行评估。该得分可以视为一种专家知识或外部情报,当历史数据不足时,通过引入大语言模型的知识,有助于冷启动时决策,具体方法见第 2.4.4 节。

式(8)表示了常规最优网络威胁狩猎方法选择机制,一旦从环境中积累了足够的数据,可以使用该策略。式中,$W_{\text{common}}(A)$通过计算每个动作 $a_j\in A$ 的归一化权重来确定最佳动作。访问过的方法集的总数 $N(A)$除以方法 a_j 的选择次数$N(a_j)$,确保尚未探索的网络威胁狩猎方法有机会被选中。

在选择完最优网络威胁狩猎方法后,下一步执行 execute_action 将威胁实体 n_i 输入网络威胁狩猎方法 a_j 方法中执行,得到结果子集合 C_{t+1},进而合并父节点集合 N_t 和子节点集合 C_{t+1},得到 $t+1$ 时刻的节点集合 N_{t+1}。

如算法 1 中第 11 步所示,S_t 表示当前图的结构和状态,时刻 $t+1$ 的下一个状态 S_{t+1} 是在先前状态 S_t 中添加新扩展实体 C_{t+1} 并增加 n_i 与新产生实体连接关系的新状态,表示为 $S_{t+1}\leftarrow$update_state(S_t,n_i,a_j,C_{t+1})。

2.4.3 反向激励过程

反向激励过程是对前向选择中使用的 Q_t 和 W_t 的调整,主要包括对实体选择和方法选择的修正。

实体选择修正是一个递归过程,其结合新的信息或反馈来更新实体的质量评估 Q,将变化进行前向传播,该递归更新过程在函数 update_node_rewards 中实现。Reward(n')的值可以通过预设奖励以无监督方式确定。为了有效放大激励,遍历所有 n'的祖先节点,更新相关实体质量,如式(9)所示:

$$Q(n)\leftarrow Q(n)+\text{Reward}(n'),$$
$$\text{s.t. }\forall n'\in C_{t+1},\forall n\in\text{Ancestors}(n') \tag{9}$$

式中:Ancestors(n')表示获取新产生节点 n'的所有祖先节点集合。

方法选择的修正是动作权重 $\boldsymbol{W}_t$ 的更新过程,在函数 update_method_rewards 中实现。该矩阵的更新需要考虑到当前方法对其他方法的潜在影响。权重更新机制如式(10)所示:

$$w_{a_j}\leftarrow w_{a_j}+\text{Reward}(n')\times S_{\text{method}}(a,a_j),$$
$$\forall a\in A,\forall n'\in C_{t+1} \tag{10}$$

式中:$a\in A$ 表示所有方法;a_j 表示在实体 n_i 上选择的实际方法;$S_{\text{method}}(a,a_j)$为方法 a 与方法 a_j 之间的相似性度量。

2.4.4 基于大语言模型的冷启动模块

基于大语言模型的冷启动模块需要使用大语言模型为当前决策方法进行打分,其中涉及获取待匹配威胁实体,并使用第 2.3.1 节与第 2.3.2 节中得出的 T_{text} 和 T_{LLM},以及对应的方法描述作为上下文。提示词的构成简述为:

1)任务区:描述当前任务为网络威胁狩猎任务,明确任务输入为威胁实体、网络威胁狩猎方法名,根据威胁可能性进行决策。

2)方法区:给定所有网络威胁狩猎方法名称以及对应描述。

3)上下文区:输入 T_{text} 与 T_{LLM}。

4)输入区:输入待决策节点名称。

5)输出格式与思考引导区:规定输出格式为<Number> < Method name > [Score]-Brief Explanation。

而后,请求大语言模型,等待大语言模型输出后,通过正则表达式匹配出方法 a_j 的得分,表示为 LLMScore(a_j)。

3 实验

3.1 网络威胁狩猎数据集

为了模拟真实的网络威胁狩猎场景,本文构建了模拟网络威胁狩猎的数据集,其中包含 186 份威胁情报的报告。为了模拟网络威胁狩猎过程,从威胁情报中提取初始的威胁实体集合,包括 IP 地址、域名、URL、文件哈希。而后,基于本地、VirusTotal、RapidDNS 和 ThreatBook 4 个数据源,实现了 21 种自动网络威胁狩猎方法。自动狩猎方法包括域名-IP 关系查询(当前解析、历史解析、SSL 历史解析地址)、IP-域名关系查询(历史解析)、域名-域名关系查询(子域名)、文件-IP 关系查询(通信地址)、文件-文件关系查询(打包关系、子调用关系)以及文件-域名关系查询(通信地址)。创建数据集时,分别对每篇威胁情报的初始威胁实体集合进行扩展,模拟专家对某单一事件的网络威胁狩猎过程。在扩展过程中,记录不同数据源对实体是否标记为恶意节点,作为下文评测效率的核心指标。

对产生的数据集进行数据分析,每份报告平均包含 11.77 个初始威胁实体,经过扩展后,平均实体

www.ecice06.com

数量增加 415.80 个，平均关系数量为 925.47 个。将关键关系定义为连接威胁实体的唯一关系，对关系数量以及关键关系数量进行统计，统计结果如表 1 所示，其中，最优值加粗标示，次优值以下划线标示。不同数据源占所有关键关系的贡献比例如下：VirusTotal 占 34.7%，本地占 34.0%，RapidDNS 占 23.3%，ThreatBook 占 8.0%。对数据集的分析表明，不同数据占比较为均衡，可以有效模拟真实场景下的网络威胁狩猎场景。

表 1 数据集中关系数量、关键关系数量及占比

Table 1 Number of relationships, key relationships, and proportions in the dataset

数据源名称	关系数量及占比	关键关系数量及占比
本地	**71 760 (41.7%)**	197 (34.0%)
VirusTotal[38]	27 193 (15.8%)	**201 (34.7%)**
ThreatBook[39]	20 881 (12.1%)	47 (8.0%)
RapidDNS[40]	52 303 (30.4%)	135 (23.3%)

3.2 评估指标

为了评估 RE-HUNTER 方法的有效性，定义了以下指标：

1)平均召回率。该指标用于评估在网络威胁狩猎过程中已发现的恶意威胁实体占所有恶意威胁实体的比例，具体定义为每个事件中，恶意威胁实体发现率的平均值，如式(11)所示：

$$R=\frac{1}{N}\sum_{i=1}^{N}\frac{N_{\mathrm{TP}i}}{N_{\mathrm{TP}i}+N_{\mathrm{FN}i}} \tag{11}$$

式中：$N_{\mathrm{TP}i}$ 表示第 i 个网络威胁狩猎过程中发现的恶意威胁实体数量；$N_{\mathrm{FN}i}$ 表示未发现的恶意威胁实体数量；N 表示网络威胁狩猎事件总数。该指标适用于评估在运行指定步数时，不同算法对网络威胁狩猎任务的有效性和贡献。

2)达到召回率的平均步数。该指标量化了每个事件中达到特定召回率阈值所需的平均决策步数，计算过程如式(12)所示：

$$S_{\mathrm{R}}=\frac{1}{N}\sum_{i=1}^{N}s_{r_i} \tag{12}$$

式中：s_{r_i} 是第 i 个网络威胁狩猎事件中，决策算法达到指定召回率阈值所需的步数。该指标适用于评估在网络威胁狩猎任务中实现不同水平的搜索完整性所需的成本。

3)平均累计奖励。该指标衡量网络威胁狩猎过程中算法获得的平均累积奖励值，反映了算法在整个决策过程中的综合性能，计算过程如式(13)所示：

$$J=\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}r_i(t) \tag{13}$$

式中：$r_i(t)$ 表示第 i 个网络威胁狩猎事件中第 t 步获得的奖励；T_i 表示第 i 个事件的总步数。

3.3 实验设置

3.3.1 方法设置

本文进行以下实验项目：

1)网络威胁狩猎方法实验：对本文提出的方法 RE-HUNTER 与其他方法进行性能对比。

2)消融实验：对本文方法的不同组成模块进行消融，验证不同模块的有效性，并验证不同大语言模型对方法效果的影响。

3)奖励策略实验：验证不同奖励策略对方法性能的影响，主要对比方法如表 2 所示。在开始网络威胁狩猎前，将随机数生成器种子设置为固定，以确保多次实验结果的一致性。

3.3.2 奖励策略与超参数设置

为了解决稀疏性，本文采用 AllRewardPolicy 作为默认奖励策略。此外，对以下奖励策略进行实验以验证不同奖励策略的性能：

AllRewardPolicy：对于每个新产生的实体，无论是否已经出现在当前网络威胁狩猎决策过程中，增加 1 点奖励，设计意图是缓解稀疏性。

NewResultRewardPolicy：对于每个新产生的实体，若从未出现在网络威胁狩猎决策过程中则增加 1 点奖励，设计意图是奖励能够进一步扩展图的方法。

MalwareRewardPolicy：对新产生实体的恶意性质进行查询，若查询为恶意，则增加 1 点奖励。

IoCPathRewardPolicy：每当在初始威胁实体集合之间形成路径时增加 1 点奖励，设计意图是尝试将自动化方法与文本进行对齐，尝试对专家历史经验进行模拟。

评测过程中每种方法使用不同超参数进行实验：0.000 01、0.000 1、0.001、0.01、0.1、1、5、7、10、20、50、100 以及 1 000。最终结果取每种方法的最佳状态，最佳状态通过选择在所有奖励策略和超参数设置中产生最高平均召回率的配置进行确定。

3.4 实验结果

3.4.1 网络威胁狩猎方法实验结果

如表 3 所示，总体而言，本文提出的方法在不同召回率下，均实现了最少的操作步数。在 10%～30%召回率的初始阶段，显著减少了所需的操作步数：在 10%的召回率下，相比最优基线方法减少了 63.41%的操作步数；在 20%时，所需操作步数减少了 53.85%。中长期条件下，30%～60%的召回率所需操作步数平均减少了 13.47%。对于 70%～90%的固定召回率，所需操作步数平均减少了 27.56%。

计算机工程 www.ecice06.com

表 2 不同实验方法、来源及描述比较

Table 2 Comparison of different experimental methods, sources, and descriptions

方法名称	来源	描述
SA-UCB	文献[8]	基于多臂赌博机进行建模，实体选择时，选取在不同攻击事件中出现频率最高的实体，并使用 UCB 算法进行方法选择；为了计算威胁实体在其他攻击事件中的出现频率，此方法可直接访问数据集中所有的文献与图结构、节点信息
SA-UCB-REFINE	文献[8]	基于分层马尔可夫决策过程进行建模，改进 SA-UCB 方法，通过引入 UCB 策略作为实体选择出现频次较少时的替代措施，缓解数据稀疏带来的影响
MCTS-UCB	文献[11]	基于分层马尔可夫决策过程进行建模，基于 MCTS，使用 UCB 策略分别对威胁实体和狩猎方法进行选择
MCTS-EGreedy	文献[26]	基于分层马尔可夫决策过程进行建模，基于 MCTS，使用 ε-greedy 策略分别对威胁实体和狩猎方法进行选择
MCTS-UCB-RT	消融方法	基于 MCTS-UCB 方法，对第 2.4.1 节实体选择过程、第 2.4.3 节反向激励过程中的递归更新机制进行改进
MCTS-UCB-RT-WT	消融方法	基于 MCTS-UCB-RT 方法，对第 2.4.2 节方法选择过程、第 2.4.3 节反向激励过程中的方法相似度机制进行改进
RE-HUNTER-WITHOUT-CONTEXT	消融方法	基于 MCTS-UCB-RT-WT，引入基于大语言模型的冷启动模块，使用大语言模型进行冷启动(GPT-3.5-turbo)，但不进行上下文注入
RE-HUNTER-[大模型名称]	消融方法	本文提出方法，使用文中提出的所有机制，分别更换基于大语言模型的冷启动模块中的大语言模型为 Llama-3.3-70B-Instruct[40]、Qwen2.5-72B-Instruct[41]、DeepSeek-R1-Distill-Qwen-32B[42]对决策效果进行观察
RE-HUNTER	本文方法	本文提出的方法，使用文中提出的所有机制(基于 GPT-3.5-turbo[43]模型)

表 3 各方法达到不同召回率所需的平均步数

Table 3 Average steps required to reach different recall rates

方法	10%召回率	20%召回率	30%召回率	40%召回率	50%召回率	60%召回率	70%召回率	80%召回率	90%召回率
RE-HUNTER	**15**	**48**	**147**	**257**	**381**	**558**	**816**	**1 220**	**1 894**
SA-UCB	286	784	1 333	1 977	2 703	3 705	4 969	7 113	10 131
SA-UCB-REFINE	63	151	261	374	573	792	1 131	1 654	2 941
MCTS-UCB	41	104	177	284	449	636	1 031	1 774	2 994
MCTS-EGreedy	183	288	394	543	750	1 017	1 464	2 166	3 675
最优方法较次优方法成本降低幅度/%	63.41	53.85	16.95	9.51	15.14	12.26	20.85	26.24	36.75

如图 2(a)所示，在 0～2 000 步的区间内，与不同的基线方法相比，RE-HUNTER 较 MCTS-UCB 平均召回率提高了 18.24%，较 MCTS-EGreedy 提高了 31.5%，较 SA-UCB-REFINE 提高了 33.10%，较 SA-UCB 提高了 351%；在 0～250 步的冷启动区间内，RE-HUNTER 较最优基线方法 MCTS-UCB 提高了 86.28%，较 SA-UCB-REFINE 提高了 143.70%，较 MCTS-EGreedy 提高了 372.92%，较 SA-UCB 提高了 1 375%。因此，在网络威胁狩猎任务的不同决策步数下，RE-HUNTER 全程表现出较高平均召回率。

对平均累计奖励进行分析，如图 2(b)所示。在 0～2 000 步区间内，RE-HUNTER 较 MCTS-UCB 平均累计奖励的相对提升为 32.68%，较 SA-UCB-REFINE 的平均相对提升为 21.08%，较 MCTS-EGreedy 的平均相对提升为 49.77%，较 SA-UCB 平均相对提升为 257.54%。平均累计奖励的实验结果表明，RE-HUNTER 能够在早期快速获得较多奖励以初始化模型，在 1 200 步后仍保持稳定的奖励增长率，具备持续优化与长期学习能力。此外，对 SA-UCB-REFINE 与 SA-UCB 的实验结果进行对比分析，其平均召回率相对提升为 78%。因此，在数据稀疏的真实世界决策过程中，使用多臂赌博机进行建模效果较差，实际应用中可通过使用多层次建模、冷启动优化等手段提升其效果。

RE-HUNTER 的对比方法实验表明，所提出的方法在真实网络威胁狩猎场景下，较不同对比方法展现出较大优势，特别地，RE-HUNTER 方法能够大幅提升冷启动条件下的决策效果。

计算机工程 www.ecice06.com

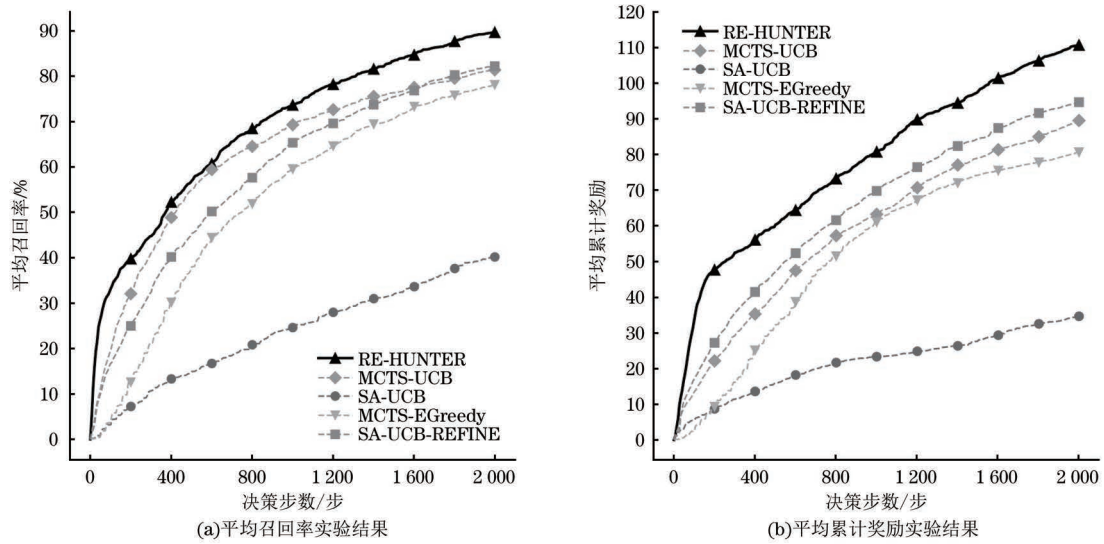


图 2 网络威胁狩猎方法对比结果

Fig. 2 Comparison of cyber threat hunting methods

3.4.2 消融实验结果

对本文提出的 RE-HUNTER 方法进行消融实验，结果如图 3(a)所示。MCTS-UCB-RT 方法与基础方法 MCTS-UCB 相比，平均召回率下降了 4.02%，但在 250～2 000 步区间，平均提高了 3.55%。因此 MCTS-UCB-RT 能够提升长期性能。

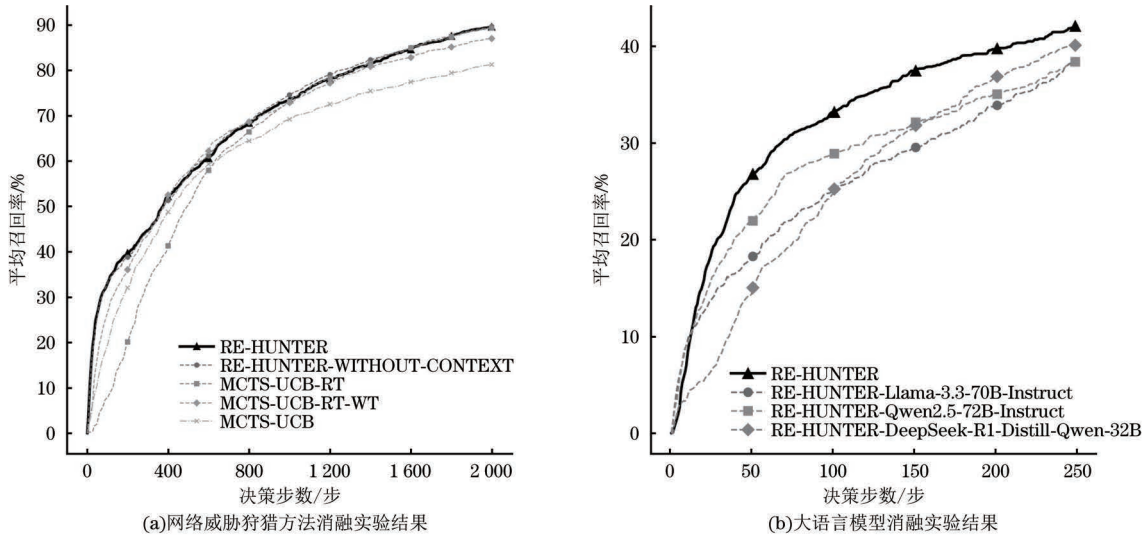


图 3 网络威胁狩猎消融实验结果

Fig. 3 Ablation experiment results for cyber threat hunting

在 MCTS-UCB-RT 的基础上添加方法质量与方法相似度机制，形成了 MCTS-UCB-RT-WT 方法，较 MCTS-UCB-RT，其平均召回率在 0～2 000 步区间提高了 9.02%，在 0～250 步区间提高了 7.22%。因此，方法质量与方法相似度机制提升了全局性能。

在 MCTS-UCB-RT-WT 的基础上引入大语言模型以及冷启动机制，但是不注入待决策实体的对应上下文，形成了 RE-HUNTER-WITHOUT-CONTEXT 方法。在 0～2 000 步区间的平均召回率比基线提高了 15.86%。具体而言，在 0～250 步区间，平均召回率大幅提升了 76.46%。与没有引入大语言模型的 MCTS-UCB-RT-WT 方法相比，在 0～2 000 步区间的平均召回率提高了 5.70%，在 0～250 步区间的平均召回率显著提高了 45.69%，而在 250～2 000 步区间没有明显提高。这表明添加大语言模型模块增强了初始启动速度，但对长期性能影响较小。

在 RE-HUNTER-WITHOUT-CONTEXT 的基础上，添加上下文模块，形成了本文提出的 RE-HUNTER 方法。与 MCTS-UCB 相比，其平均召回率在初始的 0～250 步区间提高了 86.28%，在 0～

1 000 步区间提高了 27.41%，在 0～2 000 步区间提高了 18.24%，在 250～2 000 步区间提高了 8.49%。与 RE-HUNTER-WITHOUT-CONTEXT 方法相比，添加上下文模块的 RE-HUNTER 方法在 0～2 000 步区间平均召回率提高了 1.51%，在启动阶段(0～250 步)提高了 3.68%，在 250～2 000 步区间提高了 1.20%。

将大语言模型替换为其他不同模型，实验结果表明，不同大语言模型性能差异主要体现在 0～250 步区间的平均召回率，如图 3(b)所示。其中，默认使用的 GPT-3.5-turbo 模型较 Qwen2.5-72B-Instruct 的相对提升为 14.38%，较 Llama-3.3-70B-Instruct 的相对提升为 24.52%，较 DeepSeek-R1-Distill-Qwen-32B 的相对提升为 25.53%。这说明决策性能与大语言模型性能相关，但相关开源模型效果略低于商用模型。消融实验表明 RE-HUNTER 方法结合了 MCTS-WT-RT 和 MCTS-UCB-RT 的优势，在早期和中期阶段表现出优秀的性能，而 MCTS-UCB-RT 在长期性能方面表现更好。此外，大语言模型能够有效提升冷启动阶段性能，提升效果与大语言模型能力相关。

3.4.3 奖励策略实验结果

本实验在 SA-UCB、SA-UCB-REFINE、MCTS-UCB 与 RE-HUNTER 方法中替换了不同的奖励策略。RE-HUNTER 方法的实验结果如图 4(a)所示，在冷启动阶段获得的专家知识使不同奖励策略之间的性能差距比其他方法更小。从初始阶段来看，AllRewardPolicy 实现了最佳性能，而从长期看，NewResultRewardPolicy 表现最佳。

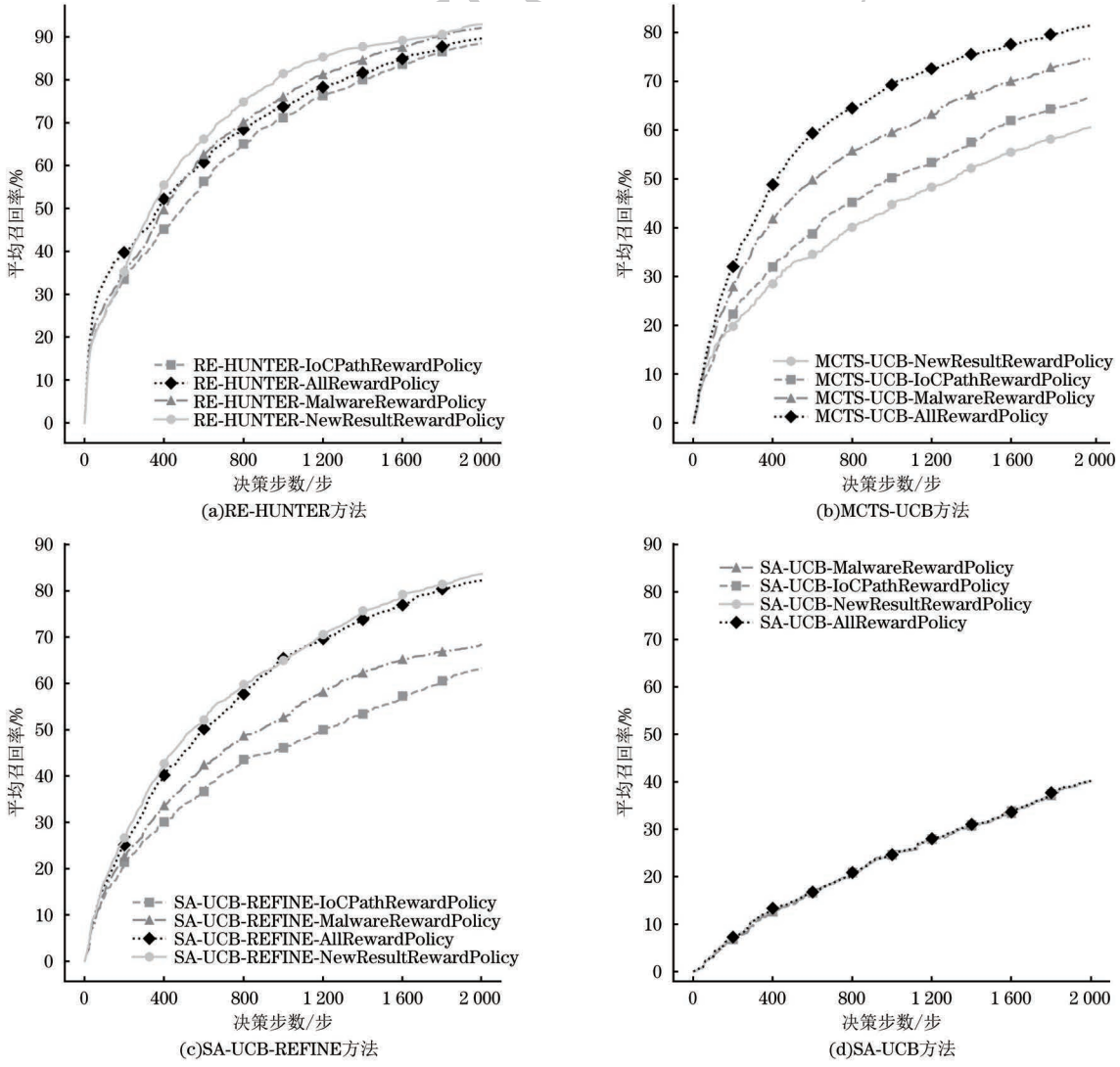


图 4 不同强化学习奖励策略在不同方法中的应用效果

Fig.4 Effectiveness of different reinforcement learning reward policies across various methods

MCTS-UCB 方法的实验效果如图 4(b)所示，使用 AllRewardPolicy 实现了最佳性能，而 MalwareReward 的效率与 AllRewardPolicy 相比较低，IoCPathReward 在早期阶段效率较低，但随后超过了 NewResultRewardPolicy。

SA-UCB-REFINE 方法的实验效果如图 4(c)所

示，AllRewardPolicy 实现了与 NewResultRewardPolicy 相似的性能。

SA-UCB 方法的实验效果如图 4(d)所示，由于该方法使用“most share”策略进行实体选择，而后根据 UCB 策略进行狩猎方法选择，其性能退化为广度优先搜索，因此所有奖励策略都表现出相同的性能。

奖励策略实验结果说明，虽然任务的目标是发现恶意威胁实体，但是在大多数方法中，因为恶意威胁稀疏，所以奖励恶意实体识别的方法并未表现最佳。在奖励不足的情况下，使用 AllRewardPolicy 或 NewResultRewardPolicy 可以有效地提升性能。因此，在进行网络威胁狩猎过程中，需要针对方法与数据特性选择合适的奖励策略。

4 结束语

现有的网络威胁狩猎方法难以有效解决现实场景中先验知识缺乏和反馈稀疏的问题。本文提出了 RE-HUNTER 方法，将强化学习与大语言模型相结合，构建了一个具有冷启动优化能力和反馈增强机制的网络威胁狩猎决策方法。本文基于 186 个真实威胁事件的全面评估表明，RE-HUNTER 方法在网络威胁狩猎场景下具有高效的恶意实体发现能力以及扩展威胁情报的能力，降低了对广泛的先验知识和专家干预的依赖，以及网络威胁狩猎方法落地的需求。此外，本文仅研究了在主动防御场景下，通过网络威胁狩猎的手段对攻击线索和威胁实体的扩展，下一步可结合攻击组织溯源，在事件中对攻击事件进行攻击归因。

参考文献

[1] IBM. What is threat hunting? [EB/OL]. [2025-02-10]. https://www.ibm.com/think/topics/threat-hunting.

[2] GAO Y L, LI X Y, PENG H, et al. HinCTI: a cyber threat intelligence modeling and identification system based on heterogeneous information network[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(2): 708-722.

[3] SATVAT K, GJOMEMO R, VENKATAKRISHNAN V N. Extractor: extracting attack behavior from threat reports[C]// Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P). Vienna, Austria: IEEE Press, 2021: 598-615.

[4] REN Y T, XIAO Y J, ZHOU Y H, et al. CSKG4APT: a cybersecurity knowledge graph for advanced persistent threat organization attribution [J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(6): 5695-5709.

[5] FARD N E, SELMIC R R, KHORASANI K. A review of techniques and policies on cybersecurity using artificial intelligence and reinforcement learning algorithms[J]. IEEE Technology and Society Magazine, 2023, 42(3): 57-68.

[6] SUFI F. An innovative GPT-based open-source intelligence using historical cyber incident reports[J]. Natural Language Processing Journal, 2024, 7: 100074.

[7] YAO Y F, DUAN J H, XU K D, et al. A survey on Large Language Model (LLM) security and privacy: the good, the bad, and the ugly[J]. High-Confidence Computing, 2024, 4(2): 100211.

[8] DEKEL L, LEYBOVICH I, ZILBERMAN P, et al. MABAT: a multi-armed bandit approach for threat-hunting [J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 477-490.

[9] MILAJERDI S M, ESHETE B, GJOMEMO R, et al. POIROT: aligning attack behavior with kernel audit records for cyber threat hunting[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York, USA: ACM Press, 2019: 1795-1812.

[10] NADEEM A, VERWER S, MOSKAL S, et al. Alert-driven attack graph generation using S-PDFA [J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 731-746.

[11] KOCSIS L, SZEPESVÁRI C. Bandit based Monte-Carlo planning [C] // Proceedings of European Conference on Machine Learning. Berlin, Germany: Springer, 2006: 282-293.

[12] ABU M S, SELAMAT S R, ARIFFIN A, et al. Cyber threat intelligence—issues and challenges [J]. Indonesian Journal of Electrical Engineering and Computer Science, 2018, 10(1): 371-379.

[13] WAGNER C, DULAUNOY A, WAGENER G, et al. MISP: the design and implementation of a collaborative threat intelligence sharing platform[C]//Proceedings of the 2016 ACM Workshop on Information Sharing and Collaborative Security. New York, USA: ACM Press, 2016: 49-56.

[14] WAGNER T D, MAHBUB K, PALOMAR E, et al. Cyber threat intelligence sharing: survey and research directions [J]. Computers & Security, 2019, 87: 101589.

[15] 杨秀璋, 彭国军, 刘思德, 等. 面向 APT 攻击的溯源和推理研究综述[J]. 软件学报, 2025, 36(1): 203-252.
YANG X Z, PENG G J, LIU S D, et al. Survey on attribution and inference research for APT attacks [J]. Journal of Software, 2025, 36(1): 203-252. (in Chinese)

[16] MILAJERDI S M, GJOMEMO R, ESHETE B, et al. HOLMES: real-time APT detection through correlation of suspicious information flows[C]//Proceedings of the IEEE Symposium on Security and Privacy (SP). San Francisco, USA: IEEE Press, 2019: 1137-1152.

[17] GAO P, SHAO F, LIU X Y, et al. Enabling efficient cyber threat hunting with cyber threat intelligence[C]//Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE). Chania, Greece: IEEE Press, 2021: 193-204.

[18] ALSAHEEL A, NAN Y H, MA S Q, et al. ATLAS: a sequence-based learning approach for attack investigation[C]// Proceedings of the USENIX Security Symposium. [S. l.]: USENIX Association, 2021: 3005-3022.

[19] SUN N, DING M, JIANG J J, et al. Cyber threat intelligence mining for proactive cybersecurity defense: a survey and new perspectives [J]. IEEE Communications Surveys & Tutorials, 2023, 25(3): 1748-1774.

[20] 崔琳, 杨黎斌, 何清林, 等. 基于开源信息平台的威胁情报挖掘综述[J]. 信息安全学报, 2022, 7(1): 1-26.
CUI L, YANG L B, HE Q L, et al. Survey of cyber threat intelligence mining based on open source information platform [J]. Journal of Cyber Security, 2022, 7(1): 1-26. (in Chinese)

[21] 李沁东, 陈兴蜀, 唐文佚. 开源威胁情报生产与应用综述[J]. 网络空间安全科学学报, 2023, 1(1): 59-80.

计算机工程 www.ecice06.com

LI Q D, CHEN X S, TANG W Y. A survey of open-source threat intelligence production and application[J]. Journal of Cybersecurity, 2023, 1(1): 59-80. (in Chinese)
[22] SHAKYA A K, PILLAI G, CHAKRABARTY S. Reinforcement learning algorithms: a brief survey[J]. Expert Systems with Applications, 2023, 231: 120495.
[23] ZHONG L F, WU J, LI Q, et al. A comprehensive survey on automatic knowledge graph construction[J]. ACM Computing Surveys, 2024, 56(4): 1-62.
[24] SHEN Y, CHEN J, HUANG P S, et al. M-walk: learning to walk over graphs using Monte Carlo tree search[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18). Red Hook, USA: Curran Associates Inc., 2018: 6787-6798.
[25] XIONG W H, HOANG T, WANG W Y. DeepPath: a reinforcement learning method for knowledge graph reasoning[EB/OL]. (2018-07-07)[2025-02-10]. https://arxiv.org/abs/1707.06690.
[26] VODOPIVEC T, SAMOTHRAKIS S, STER B. On Monte Carlo tree search and reinforcement learning[J]. Journal of Artificial Intelligence Research, 2017, 60: 881-936.
[27] CHANG Y P, WANG X, WANG J D, et al. A survey on evaluation of large language models[J]. ACM Transactions on Intelligent Systems and Technology, 2024, 15(3): 1-45.
[28] ZHAO Z R, LEE W S, HSU D. Large language models as commonsense knowledge for large-scale task planning[EB/OL]. [2025-02-10]. https://arxiv.org/abs/2305.14078.
[29] HU B, ZHAO C Y, ZHANG P, et al. Enabling intelligent interactions between an agent and an LLM: a reinforcement learning approach[EB/OL]. (2024-06-21)[2025-02-10]. https://arxiv.org/abs/2306.03604.
[30] SANNER S, BALOG K, RADLINSKI F, et al. Large language models are competitive near cold-start recommenders for language-and item-based preferences[C]// Proceedings of the 17th ACM Conference on Recommender Systems. New York, USA: ACM Press, 2023: 890-896.
[31] HOU Y P, ZHANG J J, LIN Z H, et al. Large language models are zero-shot rankers for recommender systems[C]// Proceedings of European Conference on Information Retrieval. Berlin, Germany: Springer, 2024: 364-381.
[32] HE Z K, XIE Z H, JHA R, et al. Large language models as zero-shot conversational recommenders[C]//Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2023: 720-730.
[33] GAO Y F, XIONG Y, GAO X Y, et al. Retrieval-augmented generation for large language models: a survey[EB/OL]. (2023-12-18)[2025-02-10]. https://arxiv.org/abs/2312.10997.
[34] MANI S K, ZHOU Y J, HSIEH K, et al. Enhancing network management using code generated by large language models[C]//Proceedings of the 22nd ACM Workshop on Hot Topics in Networks. New York, USA: ACM Press, 2023: 196-204.
[35] MENG X Y, LIN C G, WANG Y Q, et al. NetGPT: Generative pretrained transformer for network traffic[EB/OL]. (2023-04-19)[2025-02-10]. https://arxiv.org/abs/2304.09513.
[36] ALI T, KOSTAKOS P. HuntGPT: integrating machine learning-based anomaly detection and explainable AI with large language models (LLMs)[EB/OL]. (2023-09-27)[2025-02-10]. https://arxiv.org/abs/2309.16021.
[37] VirusTotal. VirusTotal-analyze suspicious files, domains, IPs and URLs to detect malware and other breaches[EB/OL]. [2025-02-10]. https://www.virustotal.com/.
[38] ThreatBook. 微步在线X情报社区[EB/OL]. [2025-02-10]. https://x.threatbook.com/.
ThreatBook. ThreatBook X threat intelligence community[EB/OL]. [2025-02-10]. https://x.threatbook.com/. (in Chinese)
[39] RapidDNS. RapidDNS—rapid DNS information collection[EB/OL]. [2024-02-10]. https://rapiddns.io/.
[40] Hugging Face. Llama-3.3-70B-Instruct[EB/OL]. [2025-02-10]. https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct/.
[41] Hugging Face. Qwen2.5-72B-Instruct[EB/OL]. [2025-02-10]. https://huggingface.co/Qwen/Qwen2.5-72B-Instruct/.
[42] Hugging Face. DeepSeek-R1-Distill-Qwen-32B[EB/OL]. [2025-02-10]. https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B/.
[43] Open AI. Model-OpenAI API[EB/OL]. [2025-02-10]. https://platform.openai.com/docs/models/gpt-3.5-turbo/.

文字编辑 金胡考
栏目编辑 赖玉玲

计算机工程 www.ecice06.com