PaperPass[旗舰版]查重报告

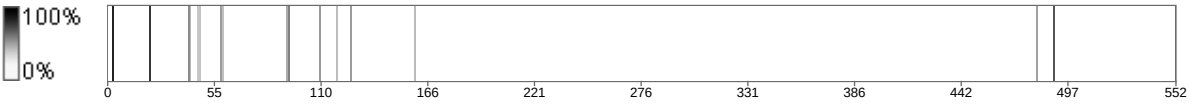
简明打印版

查重结果(相似度):

总　体：　3%
本地库：　3%　(本地库包含期刊库、学位库、会议库、联合库、图书库、报纸库、专利库、外文库)
• 期刊库：　2%　(期刊库相似度是指论文与学术期刊库的比对结果)
• 学位库：　2%　(学位库相似度是指论文与学位论文库的比对结果)
• 会议库：　0%　(会议库相似度是指论文与会议论文库的比对结果)
• 联合库：　0%　(联合库相似度是指论文与大学生联合比对库的比对结果)
• 图书库：　0%　(图书库相似度是指论文与图书库的比对结果)
• 专利库：　0%　(专利库相似度是指论文与专利库的比对结果)
• 报纸库：　0%　(报纸库相似度是指论文与报纸库的比对结果)
• 外文库：　1%　(外文库相似度是指论文与外文库的比对结果)
互联网：　0%　(互联网相似度是指论文与互联网资源的比对结果)

检测版本：旗舰版(支持中文和外文)
报告编号：6540CB84F0DD137DA
论文题目：价值驱动的以太坊交易追踪排名方法
论文作者：雷鸣
论文字数：16792
段落个数：223
句子个数：552
提交时间：2023-10-31 17:40:20
比对范围：期刊库、硕博学位库、会议库、大学生联合比对库、书籍数据、专利库、报纸库、外文库、互联网资源
查询真伪：https://www.paperpass.com/check

句子相似度分布图:



本地库相似资源列表(期刊库、硕博学位库、会议库、大学生联合比对库、书籍数据、专利库、报纸库、外文库):
没有找到与本地库相似度高的资源

互联网相似资源列表:
没有找到与互联网相似度高的资源

价值驱动的以太坊交易追踪排名方法

雷鸣[1,2] 林怡静[1,2] 高志鹏[1,2]

查重89%
1. 网络与交换技术国家重点实验室，北京，100876
2. 北京邮电大学，北京，100876

摘　　要： 区块链技术的匿名性与价值传递特性可能被恶意攻击者利用以实施网络钓鱼或其他欺诈行为。虽然链上数据公开透明可追溯，攻击者仍可通过设计复杂的交易链路，使资产在众多账户之间进行流转。最终，这些资产可能会被集中至某交易所账户并被提取，从而实现非法的利益获取。针对上述问题，文章面向以太坊提出了一种价值驱动的交易追踪排名方法。首先收集 12 起超过百万美元的以太坊攻击案例，获取大小为达 27G 的交易数据，构建交易数据图；然后从链上抽取代币的流动池数据，计算代币历史价格，确定交易图中各交易的权重系数；最后，提出基于价值占比的动态残差缩放机制，优化交易图结构，更加偏向主要的价值流通路径。实验结果表明，文章提出方法的召回率达到 89.24%，相较于 TTR、Haircut、APPR 等算法分别提高了 10%、17%和 37%，验证提出方法在检测欺诈账户上的高效性和准确性。

关键词 区块链；交易追踪；PageRank；代币价值；欺诈账户；

Value-Driven Ethereum Fraudulent Account Detection Method

查重83%
Ming Lei[1,2], Yijing Lin[1,2] and Zhipeng Gao[1,2]
[1]（*State Key Laboratory of Networking and Switching Technology, Beijing 100876*）
[2]（*Beijing University of Posts and Telecommunications, Beijing* 100876）

Abstract: The anonymity and value transfer features of blockchain can potentially be exploited by malicious attackers to carry out phishing or other fraudulent activities. Although on-chain data is publicly transparent and traceable, attackers can still create complex transaction chains, allowing assets to circulate among numerous accounts. Ultimately, these assets may be concentrated in a specific exchange account and withdrawn, thus achieving illegal gains. To address the aforementioned issues, this article proposes a value-driven transaction tracking and ranking method for Ethereum. It starts by collecting data from 12 Ethereum attack cases exceeding one million US dollars, acquiring a transaction dataset of up to 27GB, and constructing a transaction data graph. Subsequently, it extracts token liquidity pool data from the on-chain data, calculates token historical prices, and determines the weight coefficients for transactions in the graph. Finally, we introduces a dynamic residual scaling mechanism based on value proportion to optimize the transaction graph structure, favoring the primary value flow paths. Experimental results show that the proposed method achieves a recall rate of 89.24%, which is 10%, 17%, and 37% higher compared to the TTR, Haircut, and APPR algorithms, respectively, confirming the effectiveness and accuracy of the proposed approach in detecting fraudulent accounts.

Key words: blockchain; transaction tracking; PageRank; token value; fraudulent accounts

基金项目：北京市自然科学基金面上项目（4232029）
通讯作者：高志鹏，教授，研究方向为边缘智能与区块链。E-mail: gaozhipeng@bupt.edu.cn

1 引言

比特币为代表的加密数字货币出现挑战了传统的交易模式，加密数字货币的底层技术区块链实现了公开可追溯的匿名交易，为分布式账本带来了去中心化及数据不可篡改的特性。以太坊是基于区块链技术的第二代加密数字货币，弥补了比特币在可编程性上的局限性。它将区块链的应用场景从单一的金融领域扩展到其他多个领域，使得复杂的业务逻辑可以在链上以智能合约的形式自动、可信地执行。此外，为缓解区块链的可扩展性问题，以太坊通过 Layer 2、PoS 共识等方式升级网络，提高吞吐量以支撑智能合约的高通量交易。

根据 Dune Analysis 的数据统计，以太坊已部署的智能合约数量超过 6000 万个[1]。智能合约的可编程特性使得其能够以极简的代码来控制大量资产的流动。同时，智能合约部署在区块链上便不可修改，任何需要的更新或修正都必须通过部署新的合约并可能地利用预设的机制（如“后门”）来完成。因此，区块链的去中心化、匿名性以及智能合约的可编程与不可篡改特性，在助力了以太坊生态的繁荣的同时，也为非法和犯罪活动创造了一个有潜在风险的环境。近年来，热钱包的安全漏洞以及智能合约的技术缺陷为攻击者提供了可利用之机。攻击者常针对智能合约漏洞精心构建交易，将资产从用户或项目方的地址转移到自己控制地址。根据区块链安全研究公司 Chainalysis 在 2022 年度的报告中指出，链上的违法活动导致了高达 200 亿美元的经济损失[2]。

为应对前述的安全挑战，现有解决方案主要包括以下两种策略：事前遏制和事后追踪。1）事前遏制：预防潜在的智能合约安全。在编写智能合约时，遵守智能合约编写的安全规范和最佳实践；在智能合约部署时，通过代码审计识别并修复潜在的安全漏洞。2）事后追踪：当智能合约攻击安全事件发生后，快速识别、定位并响应问题。主要利用区块链的透明性，追踪链上非法资产流动情况进行交易分析，在识别非法资产停留的账户地址后，可以通过加入代币合约黑名单，禁用账户 KYC（Know Your Customer）等方式，冻结账户资产，与其他团队共享信息，共同应对威胁。本文主要针对事后追踪策略，通过综合分析区块链上的交易数据和交易所的资产流动情况，旨在快速定位恶意账户，阻止非法资产的流通与变现。现有的研究通过引入图网络算法，对链上交易进行建模，构造出综合的交易图模型。结合图神经网络、PageRank、BFS（广度优先搜索）以及 TTR（Transaction Tracing Rank）等算法和策略，从事先知道的攻击源地址开始，逐步拓展并解析交易图，高效地锁定一连串的可疑节点，并进一步识别出可能的恶意账户。

然而，当前的算法在追踪非法代币资产的过程中存在以下问题：1）将所有种类的代币视为等价的资产，导致识别存有高价值资产的恶意账户的效率低。2）不支持识别与源节点有较长跳转距离的节点，攻击者通过频繁多跳操作可绕过算法探查，难以应对复杂场景。为了更好的追踪欺诈账户链上资产流转情况，本文收集 12 起超过百万美元的以太坊攻击案例，获取大小为达 27G 的交易数据，基于动态权重缩放机制提出改进的 TTR 算法，识别高价值恶意账户，满足复杂场景下链上资产追踪的需求，主要贡献分为以下 3 个方面。

（1）本文收集 12 起涉及超过百万美元的以太坊攻击案例，获取大小为达 27G 的交易数据，基于交易数据获取非法资产转移过程中的多种代币的相应价格，构建加密货币价格评估机制，基于该评估机制建立综合的非法交易图模型。

（2）提出基于价格优化的交易图权重分配机制，通过链上流动池获取历史区块代币价格，计算出交易流动的资产价值，并将其融入到初始化权重中，使得方法高价值流通节点获得更高的排名，以解决 TTR 算法中资产等价的问题。

（3）提出基于价值占比的动态权重残差放缩机制，通过收缩小比例价值方向和膨胀大比例价值流向的权重，使交易图整体拓展方向更倾向于大比例残差的流动方向，且延长在该方向的最大拓展距离，以解决 TTR 算法中最大跳转的问题。

2 相关工作

2.1 区块链网络图构建方式

区块链网络通常被构建为图网络，再经由图算法进行特定的下游任务。大多数研究将区块链网络构建为地址图，将账户地址构建为节点，交易构建为边。不同于传统的图网络，区块链网络中交易具备复杂的属性和特征，简单将其抽象为有向边会丢失大量信息。文献[3]根据两个地址之间的交易数量来分配分配权重，将每对账户在给定时间范围内发生的交易数量作为边的权重。文献[4]提出了使用交易图、地址图和资金流动图三种类型的图来提取比特币和以太坊网络中交易的相关特征。通过构建这些图，可以更有效地识别和分析异常交易模式。文献[5]将以太坊交易网络建模为一个时间变化图。在该图中，除了表示交易的边之外，还引入了时间边缘，用于表示节点（即以太坊地址）在时间上的变化。通过引入时间边缘，可以在同一个模型中表示交易和时间，在分析网络的动态行为和用户行为的同时考虑时间因素。

2.2 交易追踪排名算法

区块链交易追踪排名算法是构建区块链网络的下游任务之一[6]。通过分析链上数据，可以挖掘交易和代币转移的模式，进而检测欺诈账户。比特币网络基于 UTXO 进行交易[7]，网络的复杂度较低，执行此类任务通常能取得更好地效果。文献[8]通过分析比特币交易历史数据，构建了交易网络和用户网络，通过分析这两个网络的结构和外部信息，追踪涉嫌盗窃比特币的用户。文献[9]通过将比特币地址划分为不同的实体组，随后通过广度优先搜索（BFS）算法来确定比特币流动的最可能方向。文献[10]提出了一种基于图卷积网络和多层感知器的深度学习方法，其使用 Elliptic 提供的标记交易数据集进行实验，通过特征工程和数据预处理，训练了一可以检测比特币中非法交易的深度学习模型。

在账户型区块链网络中，智能合约的存在引入了用户到合约、合约到用户和合约到合约之间的交互，使得图数据的构建变得更加复杂[4]，在比特币网络中所使用的交易追踪排名方法很难被直接用于以太坊、币安链这类主要依靠智能合约交易的账户型网络。文献[11]通过图表示学习将节点向量进行聚类，并基于聚类结果提出了一种基于聚类结果和已知身份节点的恶意用户检测方法。文献[12] 提出了基于时空特征的恶意账户检测方法，包括时空爆发、度爆发、余额爆发、气价爆发和吸引力等特征，随后使用机器学习算法对这些特征进行分类，以识别恶意账户。文献[13]将 PageRank 方法应用于区块链中，并针对区块链特性，对 SWAP 交易进行了识别，并在计算节点 RANK 和 PUSH 残差时考虑代币类型、交易时间、交易方向等因素。

尽管上述方法在追踪非法资产时考虑了交易的时间、数量和代币类型，但未考虑不同代币间的价格差异。在加密货币领域，代币的价格显著地影响了资产的实际价值。因此，具有不同价值的交易对于恶意账户的检测所做出的贡献程度是有所差异的。在较为简单的场景中，如币种单一、账户交易量有限时，这些方法通常能够展现出较高的准确率。但当面对交易环境复杂化时，它们往往难以全面地识别目标恶意账户。

为了更为精准地应对这种复杂性，本文提出了一种创新方法。在传统的 TTR 基础上，引入了价格权重因子和动态权重放缩策略。这种策略的加入不仅使得代币的真实价值得到了更为精确的体现，还在复杂的交易环境下展现出了优异的性能。

3 问题定义

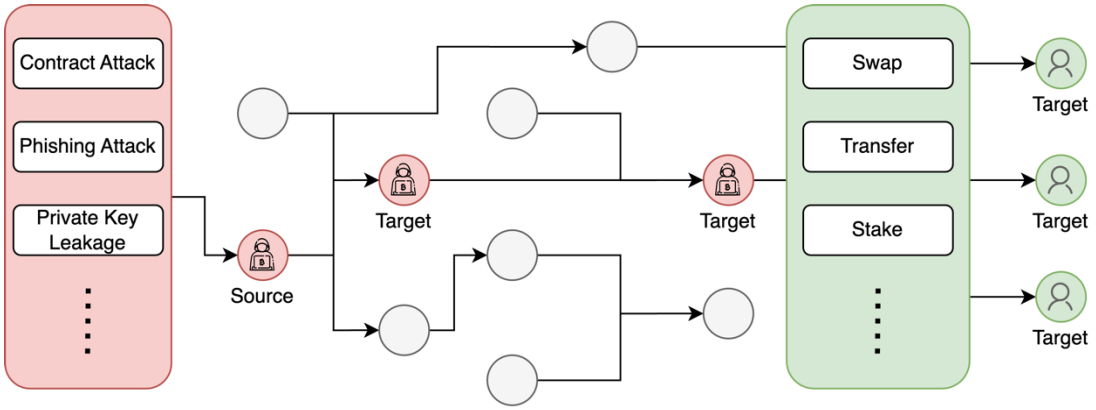


图 1 区块链攻击示意图
Figure 1 Blockchain attack workflow

在基于账户的区块链网络，如以太坊，用户可以利用智能合约来创建各种类型的代币资产。这些代币的相关行为，例如查询余额、进行转账等，都是由代币的智能合约来控制和管理的。为了规范化代币的发行和交易，以太坊社区推出了 ERC20 智能合约标准。这个标准定义了一套统一的接口规范，确保 ERC20 代币能够在不同的去中心化应用（DApp）之间实现高效的互操作性。在链上，ERC20 代币可通过去中心化金融应用（DeFi）实现相互之间的兑换。为了支持这些兑换，DeFi 平台为每个代币交易对提供了专门的智能合约。用户可以为这些交易对提供流动性，即他们可以存入两种相关的代币到一个特定的流动性池中。这里的“流动性”描述的是一个流动性池中可用于交易的资金量，且这些资金可以被迅速而不显著影响交易价格地进行交易。只要交易对的流动性池中有足够的资金，用户便可以随时进行代币间的兑换。这种机制使得去中心化交易更为流畅，降低了滑点，提高了用户的交易体验。

智能合约的代币交易生态由于其固有的可编程性、复杂性和区块链网络的匿名特性，增加了其面临的安全风险。与直接攻击区块链基础设施（如 51%攻击和双花攻击）不同，链上攻击通常在区块链正常运行的背景下展开。这些攻击方式主要包括：1）合约漏洞攻击：此类攻击与传统网络安全漏洞利用有相似之处。攻击者会发现并利用智能合约的代码缺陷，执行恶意操作来获得非法收益。其中的具体形式包括但不限于重入攻击、整数下溢攻击等。2）钓鱼攻击：这是一个广泛存在的网络诈骗方式，其在区块链领域的表现是攻击者通过创建伪造或模仿真实的区块链项目或钱包，诱使用户泄露其私钥或助记词。3）RUG PULL：这通常发生在某些区块链项目初期。项目发起者在代币价格被推高后，突然将其从流动性池中撤出，导致代币价值骤降，而项目方则利用这一点进行快速获利。

区块链技术的匿名性结合交易的高度复杂性，增加了链上资产追踪的困难度，进而对准确确定账户的真实所有者带来挑战。图 1 为区块链攻击示意图，恶意账户在通过钓鱼攻击或合约漏洞等攻击方式，获取非法加密资产后，通过与智能合约交互，非法加密资产经过流转隐藏踪迹，最终转移到伪装成正常账户的地址，随后等待进一步的出金或者其他变现操作。具体操作包括将非法资产通过混币器、Swap、Transfer 等方式进行不断地转移：1）混币器：起初是为了保护用户隐私而设计。它通过混合多个用户的资金输入，并重新为各用户分配输出，以掩盖资金的来源和去向。为应对此工具可能带来的非法资产流转，美国财政部的海外资产控制办公室（OFAC）在 2022 年对以太坊混币器 TONADO CASH 采取了一系列制裁措施，如禁止 RPC 节点的服务，限制交易所接受其资金输出等[14]。2）Transfer：这是一个标准的代币转账操作，使资产从一个账户流到另一个账户。尽管它在操作上较为直接，但在某些情境下，这种简单的交易模式仍难以追踪。例如，一个账户在短时间内先后进行了多次资金进出操作，这使得确定资金的真实来源和去向变得困难。3）Swap：这是一种通过特定交易对将一种代币转换为另一种代币的操作。但 Swap 的实现模式并不唯一，其多样性使得简单地识别和追踪成为一项挑战。

由于存在各种复杂的交易方式，如混币、转账和兑换等，欺诈行为检测在这样的环境中

变得尤为关键。以下是针对交易追踪排名任务的目标和定义：1）任务目标：从指定的源账户（Source）出发，系统地识别并追踪所有涉及非法资产流转的目标账户（Target）。核心目的是在尽可能狭窄的账户范围内，准确地锁定更多涉及的目标账户。2）源账户（Source）：此账户是欺诈活动的发起点或起始点，指的是在一次确定的非法活动中主动发起或参与的账户。3）目标账户（Target）：与源账户相对，这些是在欺诈活动中接受或参与非法资产流转的其他账户。它们可能是非法资产的中转站或最终目的地。这包括资产的临时存储账户和长期持有账户。

4 价值驱动的交易追踪排名算法

4.1 加密货币价格评估机制

图 2 为 DeFi 工作原理示意图，主要包括工厂合约和配对合约。工厂合约在 DeFi 生态中扮演了注册中心的角色，它负责为每一个加密货币交易对生成一个特定的配对合约，从而实现加密货币之间的兑换。配对合约主要负责管理资金池资金，并为用户提供兑换服务。每个配对合约构建了一个流动资金池，这个池中存储了两种代币，但数量可能不同。用户通过与配对合约互动，可以使用一种代币（代币 A）换取另一种代币（代币 B）。这个兑换的比例是由配对合约内部的算法计算得到的。如图 2 所示，DeFi 项目的工厂合约为多个交易对分别生成了配对合约，该合约为对应的交易对构建了一个流动资金池，存储了不同数量的两种代币。通过调用该智能合约，用户可以使用一定比例的代币 A 换取一个比例的代币 B，兑换比例由配对合约进行计算，在不同 DeFi 项目中存在不同的算法，但在供需关系和套利者的调整下，不同链上合约以及中心化交易所中的兑换比例，最终会趋近于相同，这个比例被认作该代币的共识比例，从而保证了价格的稳定性。

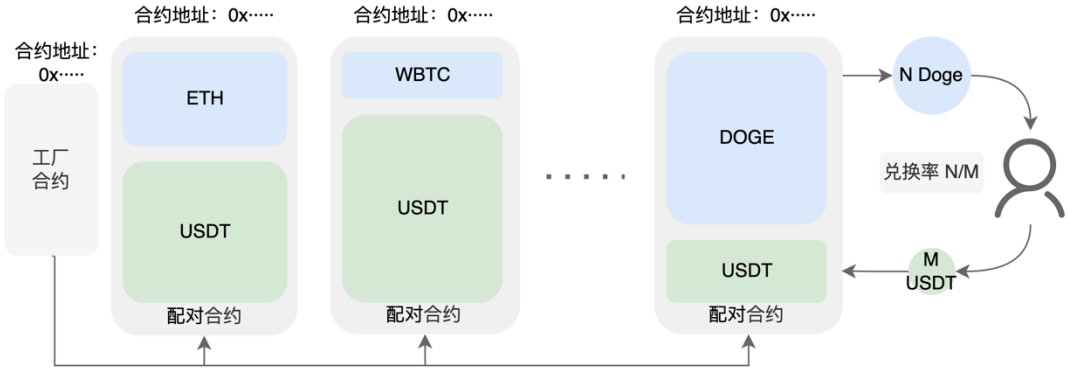


图 2 DeFi 工作原理示意图
Figure 2 DeFi workflow

为了精确地区分不同代币的价值，从而提高检测存有高价值资产恶意账户的准确性，本文调用以太坊链上 DeFi 项目 Uniswap 提供的交易对历史区块数据，从中提取代币的兑换比例以计算其价格。为确保价格的一致性并便于不同代币间的价值比对，本文将所有代币的价格标准化为以 USDT 为单位的等价价值。具体地说，一个代币与 USDT 的兑换比例被定义为其价格。相关公式如下：

$$Price(b,h) = \text{ExchangeRate}\big((b, USDT), h\big) \tag{1}$$

其中，b表示代币类型，h表示历史区块高度。对于不存在直接交易对的代币，采用路由的方式，将多个交易对的兑换比例结合计算，间接获取代币的 USDT 价格，公式为：

$$Price(b,h) = \text{ExchangeRate}\big((b, x_i), h\big) \times \cdots \times \text{ExchangeRate}\big((x_j, USDT), h\big) \tag{2}$$

对于缺乏流动性的流动池，当流动性小于一定比例后，价格并不能反映真实的代币价值，

本文将这种流动池视作无效流动池，即：

$$\begin{aligned} &\text{ExchangeRate}(\text{token1}, \text{token2}) = 0 \\ &\text{when Amount}(\text{token1}) \times \text{Amout}(\text{token2}) < \sigma \end{aligned} \tag{3}$$

4.2 价格优化后的权重分配机制

本文将整个区块链网络构建为图 $G = (V, E)$，其中 V 是节点的集合，代表网络中的所有账户，E 是边的集合，代表网络中的所有交易。对于图中的任意边 $e \in E$，本文使用 $f_b(e)$ 表示交易的代币种类，$f_v(e)$ 表示该交易中代币数量，$f_t(e)$ 和 $f_h(e)$ 表示交易发生的时间戳和区块高度，$f_{src}(e)$ 和 $f_{tgt}(e)$ 表示交易的发起者和接受者。本方法目的是定位所有的可疑节点 $V_{suspect} \subseteq V$，并为可疑节点中的每个节点 $u \in V_{suspect}$ 分配一个表示其可疑程度的 RANK 值 $P(u)$，RANK 值的分配是将残差经由 PUSH 操作得到的：在方法的初始阶段，只存在一个源节点 N_s 和分配给其的初始残差值 $R(N_s)$，随后算法会持续地对每一个分配到残差值的节点执行 PUSH 操作。在对节点 u 进行 PUSH 操作时，PUSH 操作会将节点 u 分配到的残差 $R(u)$ 的一部分转移到节点 u 的 RANK 值 $P(u)$ 中，另一部分会分配给节点 u 经由边 e 连接的邻近节点。

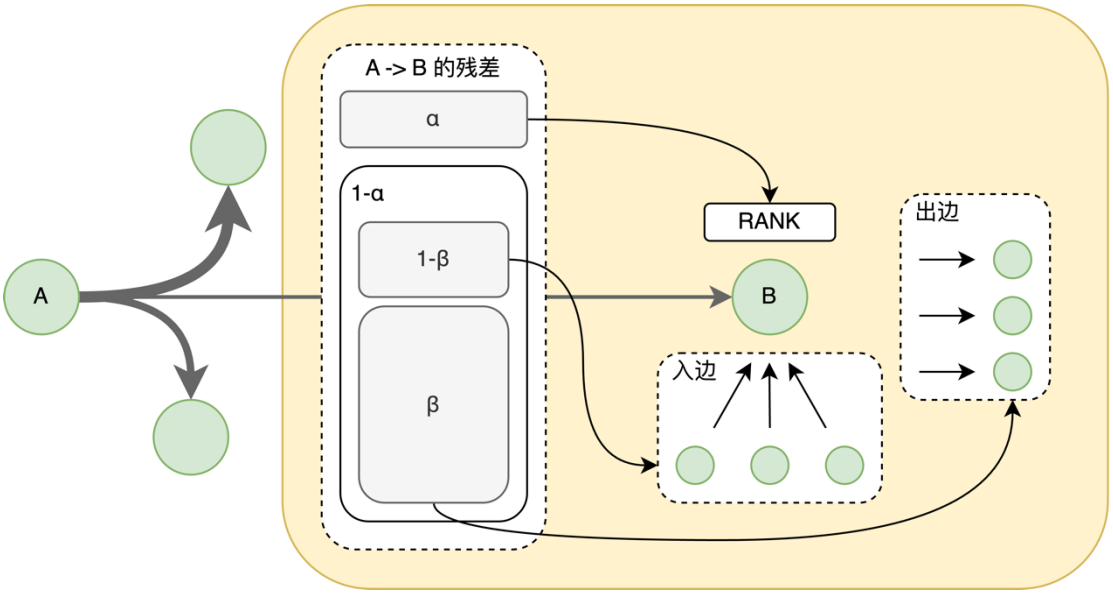


图 3 TTR 残差分配策略
Figure 3 Residual allocation strategy

如图 3 所示，在 TTR 方法的 PUSH 操作中，节点 A 分配给 B 的一笔残差主要有三个去向：1）α 比例分配给节点的 RANK 值。2）剩余 $(1-\alpha)$ 部分的 β 比例分配给出边连接的节点。3）最后的一部分 $(1-\alpha)(1-\beta)$ 分配给入边连接的节点。本文将一笔残差定义为 $r(u,t,b) \in R(u)$，t 是 e 发生的时间，b 是 e 的代币类型，$r(u,t,b)$ 表示为节点 u 在时间 t 被分配到的代币类型为 b 的一笔残差，每个节点可能经由多个边被分配到多笔残差。对于分配到节点 u 的一笔残差 $r(u,t,b)$，PUSH 阶段只会将其分配到相同代币类型边 $f_b(e) = b$ 所连接的邻居中。

本文将每种类型的代币看作一个子网络，不同代币子网络中残差的转移是分开进行的，只有当 SWAP 发生时，残差才会转移到另一个代币子网络。因此，需要在初始阶段为每个代币子网络设定总残差值。在 TTR 方法中，通过直接为源节点 N_s 的邻居分配残差，间接设置每个子网络的总残差值，对于源节点 N_s 经由边 e 连接的邻居，其通过边 e 被分配到的残差为：

$$\begin{cases} r(u,t,b) = r\left(f_{tgt}(e),\ f_t(e),\ f_b(e)\right) = \dfrac{f_v(e)}{\sum_{e_i \in E_{out}^{f_b(e)}(N_s)} f_v(e_i)}(1-\alpha)\beta & e \in E_{out}(N_s) \\ r(u,t,b) = r\left(f_{src}(e),\ f_t(e),\ f_b(e)\right) = \dfrac{f_v(e)}{\sum_{e_i \in E_{in}^{f_b(e)}(N_s)} f_v(e_i)}(1-\alpha)(1-\beta) & e \in E_{in}(N_s) \end{cases} \tag{4}$$

这里使用 $E(N_s)$ 表示与源节点 N_s 相连的所有边，根据边的方向又分为 $E_{in}(\cdot)$ 和 $E_{out}(\cdot)$，$E^b(\cdot)$ 表示代币类型为 b 的边。然而，TTR 的计算方式只关注了每种代币在链上的流通情况，而忽视了代币子网络之间追踪的价值总和的差异。如图 4 (a) 所示，TTR 方法为连接节点 B 的价值为 100USDT 的边分配的初始残差大于连接节点 C 的价值为 90ETH 的边。然而，实际上节点 C 通过其 90ETH 的连接所代表的资产价值远大于节点 B 的 100USDT。

为解决该问题，如图 4 (b) 所示，本文提出价格优化后的权重分配机制，为每种代币 b 的原始残差加入了价格权重系数 $W_B(b)$，公式如下：

$$W_B(b) = log\left(\sum_{e \in E^b(N_s)} Price\left(f_b(e), f_h(e)\right) \times f_v(e) \quad + 1\right) + 1 \tag{5-1}$$

$$r_B(u,t,b) = W_B(b) \times r(u,t,b) \tag{5-2}$$

在以太坊交易追踪排名任务中，每笔交易都由账户主动发起，一笔交易不只代表价值的流动，还代表了两个节点之间存在关联，即使是价值为 0 的交易，也反映的账户之间一定的连通关系，因此本文并未完全按价值大小设置权重，给予了 0 价值交易一定的权重。如图 4 (b) 所示，在本文提出的方法，为每条边分配了加入真实价格系数的权重，指向节点 C 的边分配到了更大的权重，与其边价值成正比例关系。

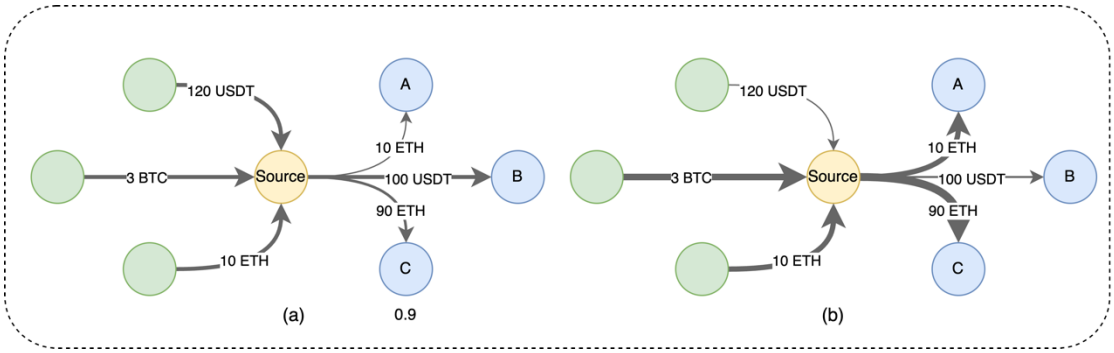


图 4 方法权重对比

Figure 4 Comparison of initial residual

4.3 基于价值占比的动态权重残差放缩机制

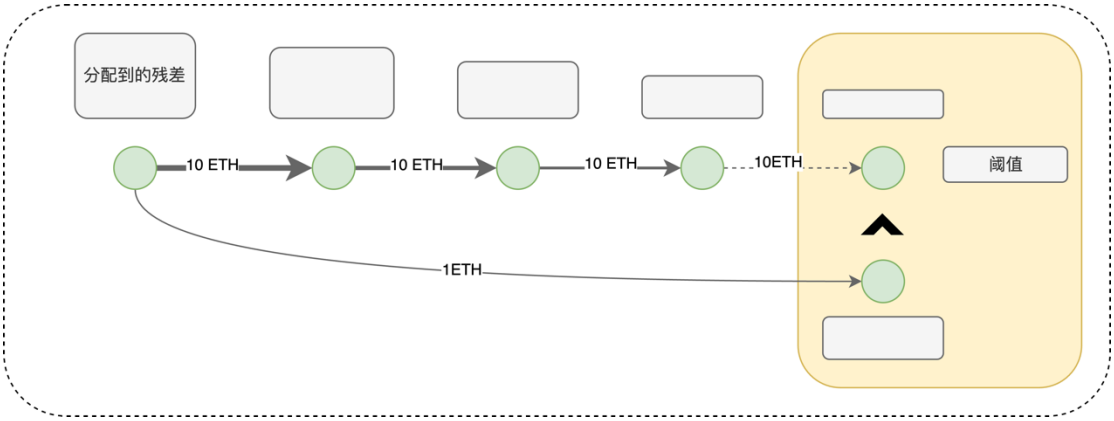


图 5 残差分配的弊端
Figure 5 Disadvantages of residual allocation

图 5 所示，原始 TTR 方法在每轮 PUSH 过程中吸收 α 比例的残差至 RANK 值，残差以指数速度缩减，随着深度的增加，快速低于阈值，造成距离源节点较远的节点无法被捕获，且获得的残差小于距离较近的低价值边连接的节点。

为了缓解这种问题，本文提出了一种基于价值比例的残差放缩机制，将占据节点较小比例的边的残差进行缩小，同时对大比例边分得的残差进行膨胀。在对节点 v 进行的 PUSH 的过程中，对于节点 u 分配给 v 的残差 r，固定将 α 比例的残差转移到节点自身的 RANK 之中，而剩余的 $(1-\alpha)$ 部分根据残差 r 占据 u 分配的所有残差和的比例，进行对应的膨胀和缩减，放缩的系数定义为 $A(r,\alpha)$：

$$A(r,\alpha)=\frac{tanh(Ratio(r)-1/2)\times\alpha}{(1-\alpha)} \tag{6}$$

在该公式下，针对大比例边进行残差膨胀后，不会超过原始残差 r，因此不会造成总残差上升，无法收敛的风险。该机制只会致使小比例边方向，拓展更少的节点，而在大比例边方向，拓展更多的节点，使方法整体更专注于捕获节点主要的价值流动方向。

4.4 完整算法流程

本文提出的方法总共分为三个阶段，初始化残差、POP 节点、PUSH 节点：

（1）初始化残差：基于价格优化后的权重分配机制为源节点的每个邻居分配初始残差，实现每种代币总残差的设置。残差的分配综合考虑了节点之间的连通性和价值传递，如 4.2 所述。

（2）POP 节点：在初始化残差阶段或每一轮 PUSH 残差阶段后，残差发生了转移，此时进入 POP 节点阶段。POP 节点阶段需要选出一个节点作为方法下一个要进行 PUSH 残差的节点，POP 节点的策略为选出所有代币子网络分配得到的残差总和最大的节点，当该节点的残差总和小于 ϵ 时，不再 POP 节点，训练结束。

（3）PUSH 节点：将节点所分得的残差，一部分转移到节点自身的 RANK 中，另一部分转移至节点的邻居。在分配给邻居前，需要根据基于价值占比的动态权重残差放缩机制对残差进行放缩后，使价值占比高的残差可以向更远的距离进行拓展，如 4.3 所述。

完整的算法参见 Algorithm1，其中 β 代表残差 r 分配给出边连接邻居的比例；α 代表分配给自身 RANK 的比例；$R(\cdot)$ 表示节点 $\cdot$ 被分配到的所有残差的集合；$E_{in}^{valid}(u,r)$ 和 $E_{out}^{valid}(u,r)$ 分别表示节点 u 满足残差 r 分配规则的入边和出边。

Algorithm 1 Local Push

Required: $u, E(u), R(u), P(u), \alpha, \beta, W_B, A, f_v, f_{src}, f_{tgt},$

Initialize Phrase:

$N_s =$ Source Node

for $e \in E_{out}(N_s)$ **do**

$u,\ t,\ b = f_{tgt}(e),\ f_t(e),\ f_b(e)$

$ratio = \frac{f_v(e)}{\sum_{e_i \in E^b_{out}(N_s)} f_v(e_i)}$

$P(u) \mathrel{+}= W_B(b) \times \text{ratio} \times \alpha$

$r_B(u,t,b) = W_B(b) \times ratio \times (1-\alpha) \times \beta$

$R(u)$ **append** $r_B(u,t,b)$

end

for $e \in E_{in}(u)$ **do**

$u,\ t,\ b = f_{tgt}(e),\ f_t(e),\ f_b(e)$

$ratio = \frac{f_v(e)}{\sum_{e_i \in E^b_{in}(N_s)} f_v(e_i)}$

$P(u) \mathrel{+}= W_B(b) \times ratio \times \alpha$

$r_B(u,t,b) = W_B(b) \times ratio \times (1-\alpha) \times (1-\beta)$

$R(u)$ **append** $r_B(u,t,b)$

end

Push Phrase:

$u =$ Node to be Pushed

for $r \in R(u)$ **do**

$P(u) \mathrel{+}= r \times \alpha$

for $e \in E^{valid}_{out}(u,r)$ **do**

$ratio = \frac{f_v(e)}{\sum_{e_i \in E^{valid}_{out}(u,r)} f_v(e_i)}$

$R\left(f_{tgt}(e)\right)$ **append** $A(\mathrm{r},\alpha) \times (1-\alpha) \times \beta \times r \times ratio$

end

for $e \in E^{valid}_{in}(u,r)$ **do**

$ratio = \frac{f_v(e)}{\sum_{e_i \in E^{valid}_{in}(u,r)} f_v(e_i)}$

$R\left(f_{src}(e)\right)$ **append** $A(\mathrm{r},\alpha) \times (1-\alpha) \times (1-\beta) \times r \times ratio$

end

end

5 实验和分析

本文提出的价值驱动的以太坊交易追踪算法使用 Python 语言进行开发，实验硬件配置为 Intel Xeon Gold 6226R 处理器，RAM 为 256G，操作系统为 Ubuntu 18.04。为验证算法有效性，本文收集 12 起超过百万美元的以太坊攻击案例，如表 1 所示。这些案例既包括单一代币类型的简单交易场景，也涉及了包含多种代币类型、交互频繁的复杂交易场景。这些案例涉及的交易数据总计达 27G。源账户地址和目标账户地址是经由知名的区块链安全公司，如 Peckshield 和 Chainalysis，经过深入分析后确定的。为获取所有相关交易数据，本文使用 Etherscan API 直接从链上提取，这些数据包括内部交易、ERC20 交易和外部交易等。同时，链上的流动池数据是通过调用 Uniswap API 获得的，确保了评估算法在真实应用场景下的效果。

表 1 案例介绍
Table 1 Case introduction

源地址	目标地址数量	源地址	目标地址数量
0x3130662aece32f05753d00a7b95c0444150bcd3c	4	0xd4e79226f1e5a7a28abb58f4704e53cd364e8d11	4
0xe0afadad1d93704761c8550f21a53de3468ba599	18	0xeb31973e0febf3e3d7058234a5ebbae1ab4b8c23	3
0x39fb0dcd13945b835d47410ae0de7181d3edf270	1	0xa9bf70a420d364e923c74448d9d817d3f2a77822	2
0x0e57edba0fccb1e388926193c873120cab961fee	2	0xc8a65fadf0e0ddaf421f28feab69bf6e2e589963	2
0xce1f4b4f17224ec6df16eeb1e3e5321c54ff6ede	3	0xa09871aeadf4994ca12f5c0b6056bbd1d343c029	813
0x4714a26e4e2e1334c80575332ec9eb043b61a2c4	2	0x068ac6ed5efc38a6266261b4486a8907fd7ea15f	6

实验选择了其他三种方法进行对比，分别为 Haircut[15]，APPR[16] 和 TTR：APPR 算法通过分析节点之间的拓扑结构，对节点与源节点的相关性进行排序，从而支持排序导向的、个性化的搜索；Haircut 是一种启发式偏差搜索方法，利用污染分析技术对风险交易进行追踪，它假设包含脏输入交易中的每个输出都受到一定程度的污染；TTR 是基于 APPR 改进的交易追踪算法，向其中加入交易倾向、加权污染、时间推理和代币重定向，为每个嫌疑节点进行了相关性排序。为了更好地比较本文提出方法与 TTR 的性能差异，实验时将每个案例的初始总残差保持一致，只保留残差之间的比例关系。具体参数配置如下：

表 2 实验方法参数
Table 1 Experimental method parameter

方法	α	β	ϵ	others
Haircut	-	-	-	$ratio > 0.1\%$
APPR	0.15	-	10^{-3}	-
TTR	0.15	0.7	10^{-3}	-
TTR-0.0001	0.15	0.7	10^{-4}	-
TTR-Alpha	0.15	0.7	10^{-3}	-

5.1 性能分析

表 3 性能表现对比
Table 2 Performance comparison

方法	Recall(%)	Nodes(k)	Runtime(h)
Haircut	51.92	11.43	0.01
APPR	62.58	0.10	0.03
TTR	79.09	0.46	0.82
TTR-0.0001	87.94	1.39	2.10
TTR-Alpha	**89.24**	0.69	1.49

表 3 展示了所提出的算法 TTR-Alpha 与 Haircut、APPR 和 TTR 在收集的 12 个案例上的性能对比。从结果可以看出，本文提出的 TTR-Alpha 方法的召回率最终达到了 89.24%，这比 TTR 提高了 10.15%，且明显超过了 Haircut 和 APPR 的召回率。其中，Haircut 虽然显示出较短的计算时间，但由于其扩展了大量的无效节点，并且召回率低，其整体效能显得较

差，不利于后续的分析工作。APPR 的计算时间和节点数量都较为理想，但召回率表现欠佳。相较于传统算法，本文提出的方法扩展了更多的节点，增加了计算时间，但这种时间开销考虑到召回率的明显提升是完全可以接受的。重要的是，尽管欺诈账户检测的目标是在较少的节点中捕获更多的目标节点，但在给定的参数和初始残差下，TTR_Alpha 相较于传统方法捕获了更多的可疑节点。当将本文提出的方法与 TTR-0.0001（即$\epsilon = 0.0001$）进行比较时，可以清晰地看到，TTR-0.0001 扩展了比 TTR-Alpha 多两倍的节点，且所需时间更长，但其召回率仍低于提出的方法。这表明，TTR-Alpha 不仅减少了追踪的账户粒度，还确保了追踪方向的准确性。需要注意的是，以上对比是基于所有目标账户权重均等的前提进行的。

5.2 Top N 结果对比

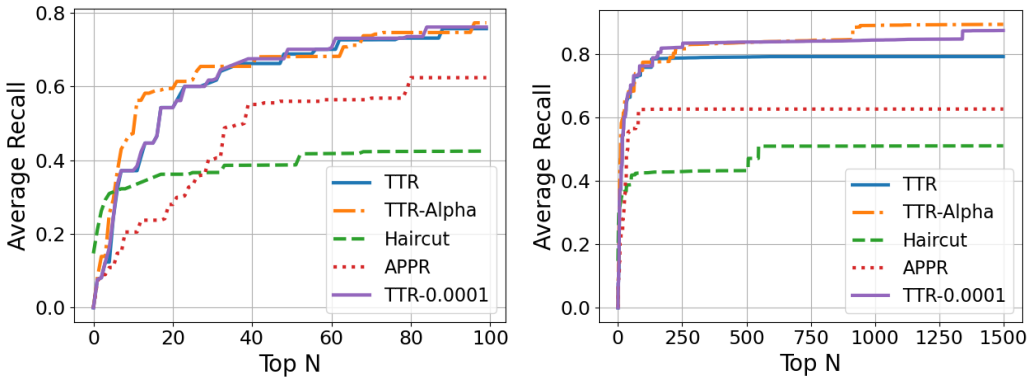


图 6 TOP N 平均召回率对比
Figure 6 Average recall of the TOP N nodes

图 6 展示了在 12 个案例中，根据 RANK 从覆盖节点中选出前 N 个时，TTR-Alpha、TTR、Haircut 和 APPR 的平均召回率表现。在前 250 个节点的分析中，本文提出的 TTR-Alpha 方法与 TTR 展现了几乎相同的召回率曲线，且它们的性能显著超过了 Haircut 和 APPR。这表明在相对简单的场景中，TTR-Alpha 和 TTR 都能迅速覆盖目标节点。但当节点数超过 250 个后，TTR-Alpha 的召回率开始稳步增长，最终比原始的 TTR 方法高出 10.15%，同时也明显优于 Haircut 和 APPR 的表现。这些发现显示，在更复杂的场景中，原始的 TTR 可能没有有效地沿着适当的方向扩展节点以覆盖更多的目标账户。相较之下，TTR-Alpha 在这种复杂情况中展现出更加卓越的性能。

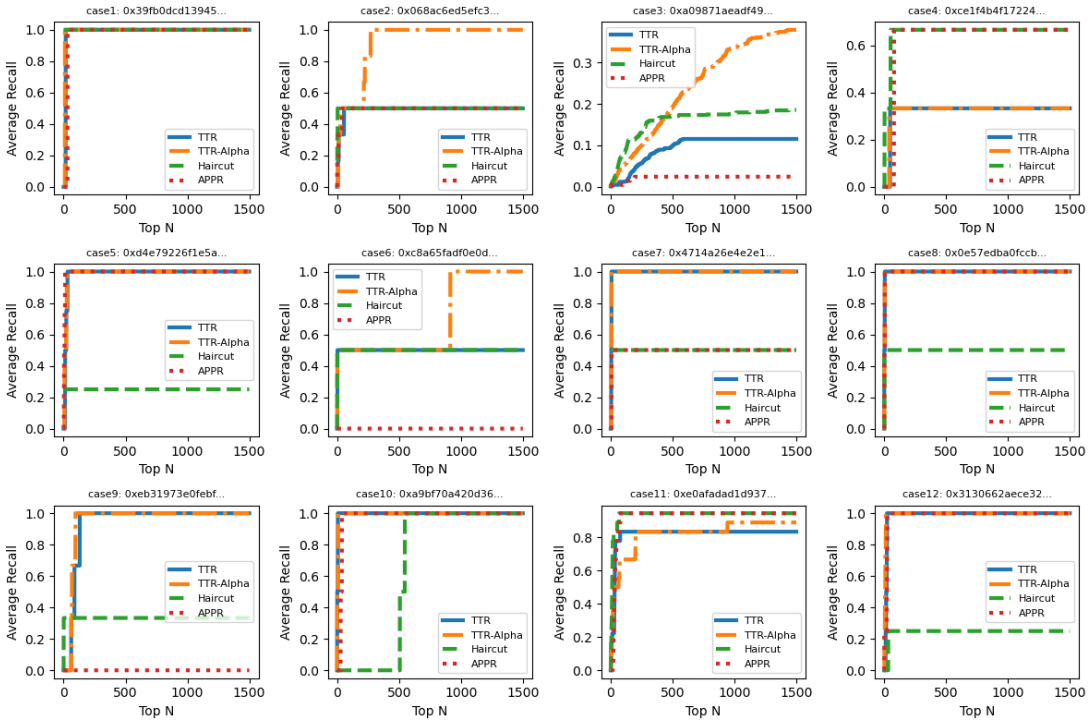


图 7 具体案例 Top N 召回率对比
Figure 7 Recall of the TOP N nodes in each case

图 7 展示了四种算法在具体案例中的 Top N 召回率表现，这些案例涵盖了从简单到复杂的不同情境。在简单的情境中，被窃取的资产由单一代币组成，而相关的账户中并没有其他可能造成干扰的交易行为。在这样的情境下，即便人工追踪这些资产所需的时间也并不会很长。因此，使用算法模型进行资产跟踪，更应关注其在复杂情境下的表现。在这 12 个案例中，本文提出的 TTR-Alpha 方法的召回率均与原始的 TTR 方法持平。尤其在复杂情境下，例如案例 2, 3, 5, 11，TTR-Alpha 相较于 TTR 能够更有效地覆盖更多的目标节点。

6 结论

本文针对现有方法在高价值账户的识别效率不足，以及对于长跳转距离节点的检测难度大的问题，提出一种新颖的价值驱动的以太坊欺诈检测策略。通过从链上提取代币的历史区块价格数据，将交易中的代币价值融合到交易图的权重中，从而在应用图算法进行欺诈账户的检测时，能更为精准地捕捉到网络中的价值传递情况。本文进一步提出一种基于价值比例的权重放缩策略，使得图算法在检测过程中更为倾向于捕获账户之间的大额价值流动动向。通过实验和分析，本文提出的策略在召回率上表现出色，并且在处理复杂交易环境时能够更为准确地定位欺诈账户。为了持续优化该方法，在未来的研究中，将对算法的时间效率进行进一步改进，并针对更为复杂的合约交易环境进行深入探索和调优。

参考文献

[1] Smart Contract Deployment Statistics[EB/OL]. (2023-05-31)[2023-10-26]. https://dune.com/pcaversaccio/smart-contract-deployment-statistics.
[2] TEAM C. 2023 Crypto Crime: Illicit Crypto Volumes Reach All-Time Highs[EB/OL]//Chainalysis. (2023-01-12)[2023-10-26]. https://www.chainalysis.com/blog/2023-crypto-crime-report-introduction/.

[3] FERRETTI S, D'ANGELO G. On the Ethereum blockchain structure: A complex networks theory perspective[J/OL]. Concurrency and Computation: Practice and Experience, 2020, 32(12): e5493. DOI:10.1002/cpe.5493.
[4] CHEN T, ZHU Y, LI Z, 等. Understanding Ethereum via Graph Analysis[C/OL]//IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. 2018: 1484-1492[2023-10-26]. https://ieeexplore.ieee.org/document/8486401. DOI:10.1109/INFOCOM.2018.8486401.
[5] ZANELATTO GAVIÃO MASCARENHAS J, ZIVIANI A, WEHMUTH K, 等. On the transaction dynamics of the Ethereum-based cryptocurrency[J/OL]. Journal of Complex Networks, 2020, 8(4): cnaa042. DOI:10.1093/comnet/cnaa042.
[6] KHAN A, AKCORA C G. Graph-based Management and Mining of Blockchain Data[C/OL]//Proceedings of the 31st ACM International Conference on Information & Knowledge Management. New York, NY, USA: Association for Computing Machinery, 2022: 5140-5143[2023-10-25]. https://dl.acm.org/doi/10.1145/3511808.3557502. DOI:10.1145/3511808.3557502.
[7] NAKAMOTO S. Bitcoin: A Peer-to-Peer Electronic Cash System[J]. 2008.
[8] REID F, HARRIGAN M. An Analysis of Anonymity in the Bitcoin System[M/OL]//ALTSHULER Y, ELOVICI Y, CREMERS A B, 等. Security and Privacy in Social Networks. New York, NY: Springer, 2013: 197-223[2023-10-25]. https://doi.org/10.1007/978-1-4614-4139-7_10. DOI:10.1007/978-1-4614-4139-7_10.
[9] ZHAO C, GUAN Y. A GRAPH-BASED INVESTIGATION OF BITCOIN TRANSACTIONS[C/OL]//PETERSON G, SHENOI S. Advances in Digital Forensics XI. Cham: Springer International Publishing, 2015: 79-95. DOI:10.1007/978-3-319-24123-4_5.
[10] A graph-based deep learning approach for illegal transaction detection in Bitcoin.[EB/OL]. (2022-10-27)[2023-10-26]. https://www.researchsquare.com. DOI:10.21203/rs.3.rs-2194869/v1.
[11] SUN H, RUAN N, LIU H. Ethereum Analysis via Node Clustering[C/OL]//LIU J K, HUANG X. Network and System Security. Cham: Springer International Publishing, 2019: 114-129. DOI:10.1007/978-3-030-36938-5_7.
[12] AGARWAL R, BARVE S, SHUKLA S K. Detecting malicious accounts in permissionless blockchains using temporal graph properties[J/OL]. Applied Network Science, 2021, 6(1): 1-30. DOI:10.1007/s41109-020-00338-3.
[13] WU Z, LIU J, WU J, 等. TRacer: Scalable Graph-Based Transaction Tracing for Account-Based Blockchain Trading Systems[J/OL]. IEEE Transactions on Information Forensics and Security, 2023, 18: 2609-2621. DOI:10.1109/TIFS.2023.3266162.
[14] U.S. Treasury Sanctions Notorious Virtual Currency Mixer Tornado Cash[EB/OL]//U.S. Department of the Treasury. (2023-10-26)[2023-10-28]. https://home.treasury.gov/news/press-releases/jy0916.
[15] MÖSER M, BÖHME R, BREUKER D. Towards Risk Scoring of Bitcoin Transactions[C/OL]//BÖHME R, BRENNER M, MOORE T, 等. Financial Cryptography and Data Security. Berlin, Heidelberg: Springer, 2014: 16-32. DOI:10.1007/978-3-662-44774-1_2.
[16] ANDERSEN R, CHUNG F, LANG K. Local Graph Partitioning using PageRank Vectors[C/OL]//2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06). Berkeley, CA, USA: IEEE, 2006: 475-486[2023-09-25]. http://ieeexplore.ieee.org/document/4031383/. DOI:10.1109/FOCS.2006.44.